

Правила создания документа JavaScript

1. Любой скрипт в HTML начинается с пары:

```
<script>  
...  
</script>
```

2. JS различает регистр.
3. Текст внутри скобок находится в кавычках, который представляет собой HTML.
4. Двойные и одинарные кавычки равноправны.
5. Внутри двойных кавычек ставятся одинарные.
6. Так как JS – объектно-ориентированный язык программирования, то при составлении скрипта будут присутствовать объекты (object), методы и свойства.

```
<script>  
document.write (“<FONT COLOR = ‘red’>Это красный текст</FONT>”)  
</script>
```

7. Вставка комментариев:

```
// в каждой строке;  
/* в начале и в конце*/
```

8. Прописывая строку JS ее необходимо уместить на одной строке редактора.
9. Каждая строка JS заканчивается «;»
10. Сочетание текста и команд требует «+» между элементами.

Типы данных в JavaScript

В JavaScript существуют следующие примитивные типы данных:

1. Число «number». Единый тип «число» используется как для целых, так и для дробных чисел.

```
var n = 123;  
n = 12.345;
```

2. Строка «string». Тип «символ» не существует, есть только «строка».

```
var str = "Мама мыла раму";  
str = 'Одинарные кавычки тоже подойдут';
```

3. Булевый (логический) тип «boolean». У него всего два значения: true (истина) и false (ложь). Как правило, такой тип используется для хранения значения типа да/нет.

```
var checked = true; // поле формы помечено галочкой  
checked = false; // поле формы не содержит галочки
```

4. Специальное значение «null». Значение null не относится ни к одному из типов выше, а образует свой отдельный тип, состоящий из единственного значения null. В JavaScript null не является «ссылкой на несуществующий объект» или «нулевым указателем», как в некоторых других языках. Это просто специальное значение, которое имеет смысл «ничего» или «значение неизвестно».

```
var age = null;
```

В частности, код говорит о том, что возраст age неизвестен.

5. Специальное значение «undefined». Значение undefined, как и null, образует свой собственный тип, состоящий из одного этого значения. Оно имеет смысл «значение не присвоено». Если переменная объявлена, но в неё ничего не записано, то её значение как раз и есть undefined. В явном виде undefined обычно не присваивают, так как это противоречит его смыслу. Для записи в переменную «пустого» или «неизвестного» значения используется null.

```
var x;  
alert( x ); // выведет "undefined"
```

Можно присвоить undefined и в явном виде, хотя это делается редко:

```
var x = 123;  
x = undefined;  
alert( x ); // "undefined"
```

6. Объекты «object», не является примитивным. Это отдельный тип. Он используется для коллекций данных и для объявления более сложных сущностей. Объявляются объекты при помощи фигурных скобок {...}.

```
var user = { name: "Вася" };
```

7. В стандарте ECMAScript 6 определен новый тип данных Symbol.

Переменные в JavaScript

Тип переменной зависит от того, какой тип информации в ней хранится. *JavaScript* не является жестко типизированным языком. Это означает, что не нужно точно определять тип данных переменной, в момент ее создания. Тип переменной присваивается автоматически в процессе выполнения скрипта.

Если во время присвоения значения переменной число заключается в кавычки, то оно рассматривается как строковое, а не числовое значение.

Можно определить переменную следующим образом:

```
let answer = 42
```

А позже, можно присвоить той же переменной следующее значение:

```
answer = "Уже текст..."
```

Так как *JavaScript* не поддерживает никаких методов и свойств для определения типа текущего значения переменной, очень важно внимательно отслеживать типы переменных во избежание неожиданных результатов.

На имя переменной в JavaScript наложены всего два ограничения.

1. Имя может состоять из: букв, цифр, символов \$ и _.
2. Первый символ не должен быть цифрой.

Целочисленные переменные могут быть десятичными (24), шестнадцатеричными (0x24) или восьмеричными (024), с плавающей запятой (5.14E6), булевы переменные (true, false).

Переменные могут быть локальными и глобальными.

Основные операторы

Сложение строк, бинарный +

Обычно при помощи плюса '+' складывают числа. Но если бинарный оператор '+' применить к строкам, то он их объединяет в одну, т.е. выполняет конкатенацию строк:

```
var a = "моя" + "строка";  
alert( a ); // моястрока
```

Если хотя бы один аргумент является строкой, то второй будет также преобразован к строке, причем не важно, справа или слева находится операнд-строка.

```
alert( '1' + 2 ); // "12"  
alert( 2 + '1' ); // "21"
```

Остальные арифметические операторы работают только с числами и всегда приводят аргументы к числу.

```
alert( 2 - '1' ); // 1  
alert( 6 / '2' ); // 3
```

Преобразование к числу, унарный плюс +

Унарный, то есть применённый к одному значению, плюс ничего не делает с числами. Однако он широко применяется, так как он позволяет преобразовать значения в число.

Например, когда полученные значения в форме строк из HTML-полей или от пользователя необходимо сложить.

```
let apples = "2";  
let oranges = "3";
```

`alert (apples + oranges);` // результат "23", так как бинарный плюс складывает строки

```
let apples = "2";  
let oranges = "3";
```

`alert (+apples + +oranges);` // 5, число, оба операнда предварительно преобразованы в числа.

Присваивание

Принцип работы такой же, как и в изученных ранее языках программирования. Возможно присваивание по цепочке:

```
let a, b, c;  
a = b = c = 2 + 2;  
alert( a ); // 4  
alert( b ); // 4  
alert( c ); // 4
```

Такое присваивание работает справа-налево, то есть сначала вычисляется самое правое выражение $2+2$, присвоится в `c`, затем выполнится `b = c` и, наконец, `a = b`.

Взятие остатка %

Его результат `a % b` – это остаток от деления `a` на `b`.

Деление на цело

Для взятия целой части от деления необходимо выполнить округление. Варианты округления:

```
num1 = 10, num2 = 3;  
Math.floor(num1/num2); // 3 (округление в меньшую сторону)  
Math.ceil(num1/num2); // 4 (округление в большую сторону)  
Math.round(num1/num2); // 3 (математическое округление)  
parseInt((num1/num2)); // 3 (приведение к числу, будет отброшена дробная часть)
```

```
<script>  
  //из числа 126 получаем 621  
  let a = 126;  
  a1 = (Math.floor(a/100));  
  a2 = (Math.floor(a/10))%10;  
  a3 = a%10;  
  alert ("Исходное число: " + a);  
  alert ("Результат: " + (a3*100 + a2*10 + a1));  
</script>
```

Инкремент/декремент: ++, --

Одной из наиболее частых операций в JavaScript, как и во многих других языках программирования, является увеличение или уменьшение переменной на единицу. Для этого существуют специальные операторы:

Инкремент ++ увеличивает на 1:

```
let i = 2;
i++;    // более короткая запись для i = i + 1.
alert(i); // 3
```

Декремент -- уменьшает на 1:

```
let i = 2;
i--;    // более короткая запись для i = i - 1.
alert(i); // 1
```

Инкремент/декремент можно применить только к переменной. Код 5++ даст ошибку.

Вызывать эти операторы можно не только после, но и перед переменной: i++ (называется «постфиксная форма») или ++i («префиксная форма»). Обе эти формы записи делают одно и то же: увеличивают на 1. Постфиксная форма i++ отличается от префиксной ++i тем, что возвращает старое значение, бывшее до увеличения.

```
// префиксная форма
let i = 1;
let a = ++i; // увеличит переменную, а затем вернёт ее значение в a
alert(a); // 2
```

```
// постфиксная форма
let i = 1;
let a = i++;
alert(a); // 1
```

Сокращённая арифметика с присваиванием

```
let n = 2;
n += 5;
n *= 2;
alert(n); // 14
```

Оператор запятая

Обычно он используется в составе более сложных конструкций, чтобы сделать несколько действий в одной строке. Например:

```
// три операции в одной строке
for (a = 1, b = 3, c = a*b; a < 10; a++) {
  ...
}
```

Работа с консолью

Консоль даёт разработчику возможность писать JavaScript-код прямо в браузере, наблюдать за тем, что происходит на страницах, и управлять этими процессами.

Чтобы в браузере открыть консоль:

Chrome – Ctrl + Shift + i

Yandex - Ctrl + Shift + j

Или в меню → Дополнительно → Дополнительные инструменты → Консоль

Методы для вывода в консоль:

`console.log(переменная)`

```
<script>
    //из числа 126 получаем 621
    let a = 126;
    a1 = (Math.floor(a/100));
    a2 = (Math.floor(a/10))%10;
    a3 = a%10;
    console.log("Исходное число: " + a);
    console.log("Результат: " + (a3*100 + a2*10 + a1));
</script>
```

`console.table(переменная)`

```
<script>

    // Я в соцсетях
    const mySocial = {
        facebook: true,
        linkedin: true,
        flickr: true,
        instagram: true,
        VKontaktebadoo: false
    };
    console.log("Я в соцсетях");
    console.table(mySocial);
</script>
```

Задание: в данный скрипт добавьте еще 2-3 соцсети с соответствующим значением

Варианты заданий

Вариант 1. Известно, что X кг конфет стоит A рублей. Определить, сколько стоит 1 кг и Y кг этих же конфет.

Вариант 2. Известно, что X кг шоколадных конфет стоит A рублей, а Y кг ирисок стоит B рублей. Определить, сколько стоит 1 кг шоколадных конфет, 1 кг ирисок, а также во сколько раз шоколадные конфеты дороже ирисок.

Вариант 3. Скорость лодки в стоячей воде V км/ч, скорость течения реки U км/ч ($U < V$). Время движения лодки по озеру T_1 ч, а по реке (против течения) — T_2 ч. Определить путь S , пройденный лодкой (путь = время • скорость). Учесть, что при движении против течения скорость лодки уменьшается на величину скорости течения.

Вариант 4. Скорость первого автомобиля V_1 км/ч, второго — V_2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили удаляются друг от друга. Данное расстояние равно сумме начального расстояния и общего пути, проделанного автомобилями; общий путь = время • суммарная скорость.

Вариант 5. Скорость первого автомобиля V_1 км/ч, второго — V_2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили первоначально движутся навстречу друг другу. Данное расстояние равно модулю разности начального расстояния и общего пути, проделанного автомобилями; общий путь = время • суммарная скорость.

Вариант 6. Дано расстояние L в сантиметрах. Используя операцию деления нацело, найти количество полных метров в нем (1 метр = 100 см).

Вариант 7. Дана масса M в килограммах. Используя операцию деления нацело, найти количество полных тонн в ней (1 тонна = 1000 кг).

Вариант 8. Дан размер файла в байтах. Используя операцию деления нацело, найти количество полных килобайтов, которые занимает данный файл (1 килобайт = 1024 байта).

Вариант 9. Даны целые положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Используя операцию деления нацело, найти количество отрезков B , размещенных на отрезке A .

Вариант 10. Даны целые положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Используя операцию взятия остатка от деления нацело, найти длину незанятой части отрезка A .

Вариант 11. Дано двузначное число. Вывести вначале его левую цифру (десятки), а затем — его правую цифру (единицы). Для нахождения десятков использовать операцию деления нацело, для нахождения единиц — операцию взятия остатка от деления.

Вариант 12. Дано двузначное число. Найти сумму и произведение его цифр.

Вариант 13. Дано двузначное число. Вывести число, полученное при перестановке цифр исходного числа.

Вариант 14. Дано трехзначное число. Используя одну операцию деления нацело, вывести первую цифру данного числа (сотни).

Вариант 15. Дано трехзначное число. Вывести вначале его последнюю цифру (единицы), а затем — его среднюю цифру (десятки).

Вариант 16. Дано трехзначное число. Найти сумму и произведение его цифр.

Вариант 17. Дано трехзначное число. Вывести число, полученное при прочтении исходного числа справа налево.

Вариант 18. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести

Вариант 19. Дано трехзначное число. В нем зачеркнули первую справа цифру и приписали ее слева. Вывести полученное число.

Вариант 20. Дано трехзначное число. Вывести число, полученное при перестановке цифр сотен и десятков исходного числа (например, 123 перейдет в 213).

Вариант 21. Дано трехзначное число. Вывести число, полученное при перестановке цифр десятков и единиц исходного числа (например, 123 перейдет в 132).

Вариант 22. Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду сотен в записи этого числа.

Вариант 23. Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду тысяч в записи этого числа.

Вариант 24. С начала суток прошло N секунд (N — целое). Найти количество полных минут, прошедших с начала суток.

Вариант 25. С начала суток прошло N секунд (N — целое). Найти количество полных часов, прошедших с начала суток.

Вариант 26. С начала суток прошло N секунд (N — целое). Найти количество секунд, прошедших с начала последней минуты.

Вариант 27. С начала суток прошло N секунд (N — целое). Найти количество секунд, прошедших с начала последнего часа.

Вариант 28. С начала суток прошло N секунд (N — целое). Найти количество полных минут, прошедших с начала последнего часа.

Вариант 29. Дни недели пронумерованы следующим образом: 0 — воскресенье, 1 — понедельник, 2 — вторник, ..., 6 — суббота. Дано целое число K , лежащее в диапазоне 1-365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было понедельником.

Вариант 30. Дни недели пронумерованы следующим образом: 0 — воскресенье, 1 — понедельник, 2 — вторник, ..., 6 — суббота. Дано целое число K , лежащее в диапазоне 1-365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было четвергом.

Вариант 31. Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 7 — воскресенье. Дано целое число K , лежащее в диапазоне 1-365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было вторником.

Вариант 32. Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 7 — воскресенье. Дано целое число K , лежащее в диапазоне 1-365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было субботой.

Вариант 33. Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 7 — воскресенье. Дано целое число K , лежащее в диапазоне 1-365, и целое число N , лежащее в диапазоне 1-7. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было днем недели с номером N .

Вариант 34. Даны целые положительные числа A , B , C . На прямоугольнике размера $A \times B$ размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике, а также площадь незанятой части прямоугольника.

<p>Вариант 1</p> <p>1. Составить программу вычисления функции: $Y=(x+1)^3+2(x-2)$.</p>	<p>Вариант 2</p> <p>1. Составить программу вычисления функции: $Y=2x^2+3(x^2-2)^2$.</p>	<p>Вариант 3</p> <p>1. Составить программу вычисления функции: $Y=2(x^2+3)+(x-3)^2$.</p>
<p>Вариант 4</p> <p>1. Составить программу вычисления функции: $Y=x^3-(x^2+1)$.</p>	<p>Вариант 5</p> <p>1. Составить программу вычисления функции: $Y=2x^2-3(x^2+1)^2$.</p>	<p>Вариант 6</p> <p>1. Составить программу вычисления функции: $Y=3(x-2)^2+2(x^2+1)$.</p>
<p>Вариант 7</p> <p>1. Составить программу вычисления функции: $Y=(x+1)^2+3(x+1)$.</p>	<p>Вариант 8</p> <p>1. Составить программу вычисления функции: $Y=6x^2+3(x^2+1)^2$.</p>	<p>Вариант 9</p> <p>1. Составить программу вычисления функции: $Y=2(x+3)+3(x+3)^2$.</p>
<p>Вариант 10</p> <p>1. Составить программу вычисления функции: $Y=x^2(x^2+1)$.</p>	<p>Вариант 11</p> <p>1. Составить программу вычисления функции: $Y=4x^2+2(x^2+1)^2$.</p>	<p>Вариант 12</p> <p>1. Составить программу вычисления функции: $Y=3(x+1)^2+2(x+1)+2$.</p>
<p>Вариант 13</p> <p>1. Составить программу вычисления функции: $Y=x^2+2(x+3)$.</p>	<p>Вариант 14</p> <p>1. Составить программу вычисления функции: $Y = \frac{x}{2} + \left(\frac{x}{2}\right)^2$</p>	<p>Вариант 15</p> <p>1. Составить программу вычисления функции: $Y = \frac{x+1}{5} + (x+1)^2$</p>
<p>Вариант 16</p> <p>1. Составить программу вычисления функции: $Y=x^2(x^2+1)$.</p>	<p>Вариант 17</p> <p>1. Составить программу вычисления функции: $Y = \frac{(x+1)^2 + 2(x+1)}{4}$</p>	<p>Вариант 18</p> <p>1. Составить программу вычисления функции: $Y = \frac{x^2}{2} + \left(\frac{x^2}{2}\right)^2 + 3$</p>
<p>Вариант 19</p> <p>1. Составить программу вычисления функции: $Y = \frac{x+1}{3} + 2(x+1)^2$</p>	<p>Вариант 20</p> <p>1. Составить программу вычисления функции: $Y = \frac{x+1}{3} + (x+1)^2$</p>	<p>Вариант 21</p> <p>1. Составить программу вычисления функции: $Y = \frac{x^2+1}{2} + \frac{27}{x^2+1}$</p>
<p>Вариант 22</p> <p>1. Составить программу вычисления функции: $Y = \frac{x-1}{5} + (x+1)^3$</p>	<p>Вариант 23</p> <p>1. Составить программу вычисления функции: $Y = \frac{x}{3} + \left(\frac{x}{2}\right)^2$</p>	<p>Вариант 24</p> <p>1. Составить программу вычисления функции: $Y = x^2 + 2x + 3$</p>

Операторы сравнения и логические значения.

Операторы сравнения следующие:

- Больше/меньше: $a > b$, $a < b$.
- Больше/меньше или равно: $a \geq b$, $a \leq b$.
- Равно $a == b$.
- Не равно $!=$.

Логические значения. Как и другие операторы, сравнение возвращает значение. Это значение имеет логический тип.

```
alert( 2 > 1 ); // true, верно  
alert( 2 == 1 ); // false, неверно  
alert( 2 != 1 ); // true
```

Логические значения можно использовать и напрямую, присваивать переменным, работать с ними как с любыми другими:

```
var a = true; // присваивать явно  
  
var b = 3 > 4; // или как результат сравнения  
alert( b ); // false  
  
alert( a == b ); // (true == false) неверно, выведет false
```

Сравнение строк. Строки сравниваются побуквенно. При этом сравниваются численные коды символов. JavaScript использует кодировку Unicode. В кодировке Unicode обычно код у строчной буквы больше, чем у прописной. Для корректного сравнения символы должны быть в одинаковом регистре.

В частности, код у символа Б больше, чем у А, поэтому и результат сравнения такой.

```
alert( 'Б' > 'А' ); // true
```

Если строка состоит из нескольких букв, то сначала сравниваются первые буквы, потом вторые, и так далее, пока одна не будет больше другой.

Если первая буква первой строки больше – значит первая строка больше, независимо от остальных символов. Если одинаковы – сравнение идёт дальше. При этом любая буква больше отсутствия буквы.

```
alert( 'Привет' > 'Прив' ); // true, так как 'е' больше чем "ничего"
```

Обычно значения, получаемые от посетителя, представлены в виде строк. Поэтому числа, полученные таким образом сравнивать нельзя, результат будет неверен.

```
alert( "2" > "14" ); // true, неверно, ведь 2 не больше 14
```

2 оказалось больше 14 , потому что строки сравниваются посимвольно, а первый символ 2 больше 1.

Правильно было бы преобразовать их к числу явным образом, поставив перед ними +

```
alert( +"2" > +"14" ); // false, теперь правильно
```

Сравнение разных типов. При сравнении значений разных типов, используется числовое преобразование. Оно применяется к обоим значениям.

```
alert( '2' > 1 ); // true, сравнивается как 2 > 1
alert( '01' == 1 ); // true, сравнивается как 1 == 1
alert( false == 0 ); // true, false становится числом 0
alert( true == 1 ); // true, так как true становится числом 1.
```

Строгое равенство. Для проверки равенства без преобразования типов используются операторы строгого равенства `===` (тройное равно) и `!==`. Если тип разный, то они всегда возвращают `false`.

```
alert( 0 === false ); // false, т.к. типы различны
```

Взаимодействие с пользователем.

Метод alert. Выводит на экран окно с сообщением и приостанавливает выполнение скрипта, пока пользователь не нажмёт «ОК». Окно сообщения, которое выводится, является модальным окном. Слово «модальное» означает, что посетитель не может взаимодействовать со страницей, нажимать другие кнопки и т.п., пока не нажмёт на «ОК».

```
alert(сообщение)
```

```
alert ( "Привет" );
```

Метод prompt выводит окно со строкой ввода.

```
Prompt ("текст в окне", "текст в строке ввода")
```

Чтобы строка ввода оставалась чистой, не заполнять вторую пару кавычек. Если вторая пара кавычек отсутствует, то в строке появится слово `undefined`. Если текст, имеющийся в строке ввода не изменить и нажать ОК, то он отобразится на странице. Если текст в строке отсутствует и ее не заполнять, то после нажатия Отмена на странице появится слово `null`.

```
<script>
/* Скрипт предназначен для того, чтобы получить
от пользователя информацию и поместить ее на страницу */

let user_name = prompt ("Напишите свое имя", "Здесь");
document.write("Привет, " + user_name + "! Милости просим!");
</script >
```

Метод confirm выводит сообщение в модальном окне с двумя кнопками: "ОК" и "ОТМЕНА".

```
<script>
    if (confirm("Сказать привет?")) {
        alert("Привет!")
    } else {
        alert("Вы нажали кнопку отмена")
    }
</script >
```

Конкретное место, где выводится модальное окно с вопросом – обычно это центр браузера, и внешний вид окна выбирает браузер. Разработчик не может на это влиять.

Задание: Создайте скрипт, который спрашивает имя и выводит его.

Условный оператор if

Условный оператор **if** имеет формат

```
If (условие)
{
    Команда(ы)
}
else
{
    Команда(ы)
}
```

Оператор **if** вычисляет и преобразует выражение в скобках к логическому типу. В логическом контексте: число 0, пустая строка "", null и undefined, а также NaN являются false, остальные значения – true.

```
if (0) { // 0 преобразуется к false, такое условие никогда не выполнится
...
}

if (1) { // 1 преобразуется к true, такое условие выполнится всегда
...
}
```

Можно передать уже готовое логическое значение, к примеру, заранее вычисленное в переменной:

```
let cond = (year != 2011); // true/false

if (cond) {
...
}
```

Пример 1

```
<script>
let year = prompt('В каком году появилась спецификация ECMA-262 5.1?', '');
if (year < 2011) {
    alert('Это слишком рано..');
} else if (year > 2011) {
    alert('Это поздновато..');
} else {
    alert('Да, точно в этом году!');
}
</script>
```

Решение задач:

1. Скрипт из **Пример 1** оформить как внешний скрипт и вызвать его на выполнение.
2. Используя конструкцию `if..else`, напишите код, который получает значение `prompt`, а затем выводит `alert`:
 - 1, если значение больше нуля,
 - 1, если значение меньше нуля,
 - 0, если значение равно нулю.

Решение:

```
let value = prompt('Введите число', 0);
if (value > 0) {
  alert( 1 );
} else if (value < 0) {
  alert( -1 );
} else {
  alert( 0 );
}
```

3. Напишите код, который будет спрашивать логин (`prompt`). Если посетитель вводит «Админ», то спрашивать пароль, если нажал отмена (`escape`) – выводить «Вход отменён», если вводит что-то другое – «Я вас не знаю».

Пароль проверять так. Если введён пароль «Чёрный Властелин», то выводить «Добро пожаловать!», иначе – «Пароль неверен», при отмене – «Вход отменён».

Решение:

```
let userName = prompt('Кто пришёл?', '');
if (userName === 'Админ') {
  var pass = prompt('Пароль?', '');
  if (pass === 'Чёрный Властелин') {
    alert( 'Добро пожаловать!' );
  } else if (pass === null) { // (*)
    alert( 'Вход отменён' );
  } else {
    alert( 'Пароль неверен' );
  }
} else if (userName === null) { // (**)
  alert( 'Вход отменён' );
} else {
  alert( 'Я вас не знаю' );
}
```

КТ № 2

<p>Вариант 1</p> <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x^3, & \text{если } X > 0 \\ 4x, & \text{если } X \leq 0 \end{cases}$ <p>3. Составить программу вычисления функции:</p> $y = \begin{cases} x , & \text{если } X < -2 \\ 2 - x, & \text{если } -2 \leq x \leq 3 \\ x, & \text{если } X > 3 \end{cases}$	<p>Вариант 2</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2, & \text{если } X < 2 \\ 2 + 3x, & \text{если } X \geq 2 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 3x, & \text{если } X < -4 \\ 2 + x, & \text{если } -4 \leq x \leq -2 \\ x - 1, & \text{если } X > -2 \end{cases}$	<p>Вариант 3</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^4, & \text{если } X > 1 \\ 4 - x, & \text{если } X \leq 1 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x + 3, & \text{если } X < -3 \\ x - 4, & \text{если } -3 \leq x \leq 3 \\ 2x, & \text{если } X > 3 \end{cases}$
<p>Вариант 4</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^3, & \text{если } X \leq 4 \\ 5x, & \text{если } X > 4 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x + 5, & \text{если } X < -5 \\ 2 - x, & \text{если } -5 \leq x \leq 1 \\ 4, & \text{если } X > 1 \end{cases}$	<p>Вариант 5</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2, & \text{если } X > -5 \\ 6x, & \text{если } X \leq -5 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 6 + x, & \text{если } X < -6 \\ \sin(x), & \text{если } -6 \leq x \leq 0 \\ 5x, & \text{если } X > 0 \end{cases}$	<p>Вариант 6</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x, & \text{если } X < 6 \\ 7x^2, & \text{если } X \geq 6 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x + 5, & \text{если } X < -7 \\ 7 - x, & \text{если } -7 \leq x \leq 3 \\ x^2, & \text{если } X > 3 \end{cases}$
<p>Вариант 7</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2, & \text{если } X > -7 \\ 7 - x, & \text{если } X \leq -7 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x - 1, & \text{если } X < -8 \\ 3x, & \text{если } -8 \leq x \leq 1 \\ 7x, & \text{если } X > 1 \end{cases}$	<p>Вариант 8</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2, & \text{если } X \leq -8 \\ 8x, & \text{если } X > -8 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 2x, & \text{если } X < -9 \\ 8 - x, & \text{если } -9 \leq x \leq 3 \\ 9, & \text{если } X > 3 \end{cases}$	<p>Вариант 9</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2 + 1, & \text{если } X > 9 \\ 2x, & \text{если } X \leq 9 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x, & \text{если } X < -1 \\ 6x, & \text{если } -1 \leq x \leq 3 \\ 10x^2, & \text{если } X > 3 \end{cases}$

<p>Вариант 10</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2 + 1, & \text{если } X > -9 \\ 4 + x, & \text{если } X \leq -9 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x+1 , & \text{если } X < -1 \\ 2x-1, & \text{если } -1 \leq x \leq 2 \\ 1 + \frac{x}{2}, & \text{если } X > 2 \end{cases}$	<p>Вариант 11</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x-1, & \text{если } X < 12 \\ 5x-2, & \text{если } X \geq 12 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 3x^2, & \text{если } X < -3 \\ 12+x, & \text{если } -3 \leq x \leq 1 \\ x -1, & \text{если } X > 1 \end{cases}$	<p>Вариант 12</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 2x^3, & \text{если } X > 11 \\ 2x-3, & \text{если } X \leq 11 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 1-x, & \text{если } X < -6 \\ \frac{x-1}{4}, & \text{если } -6 \leq x \leq -4 \\ 2x-5, & \text{если } X > -4 \end{cases}$
<p>Вариант 13</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x, & \text{если } X \leq 6 \\ \frac{4}{x}(x+1), & \text{если } X > 6 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x+1, & \text{если } X < 1 \\ \sqrt{2+x}, & \text{если } 1 \leq x \leq 2 \\ 4x, & \text{если } X > 2 \end{cases}$	<p>Вариант 14</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 1-x^2, & \text{если } X > -15 \\ \frac{4x}{3}, & \text{если } X \leq -15 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x, & \text{если } X < -1 \\ 2-x, & \text{если } -1 \leq x \leq 2 \\ 5-x , & \text{если } X > 2 \end{cases}$	<p>Вариант 15</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} \frac{x^3}{6}, & \text{если } X > -5 \\ \sqrt{4x^2+1}, & \text{если } X \leq -5 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 5-x, & \text{если } X < -4 \\ \frac{2x}{4}, & \text{если } -4 \leq x \leq 1 \\ 2x^2, & \text{если } X > 1 \end{cases}$
<p>Вариант 16</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 2x^2+3, & \text{если } X > -2 \\ 4x, & \text{если } X \leq -2 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} \sqrt{x^2+1}, & \text{если } X < -6 \\ 2+x, & \text{если } -6 \leq x \leq 2 \\ x-5, & \text{если } X > 2 \end{cases}$	<p>Вариант 17</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 1-x^2, & \text{если } X > -3 \\ 6+x, & \text{если } X \leq -3 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 2-x, & \text{если } X < 1 \\ x+2, & \text{если } 1 \leq x \leq 4 \\ 2x, & \text{если } X > 4 \end{cases}$	<p>Вариант 18</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x-1, & \text{если } X > 1 \\ 2x^2, & \text{если } X \leq 1 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 1+x, & \text{если } X < -1 \\ 6x, & \text{если } -1 \leq x \leq 2 \\ 1-x^2, & \text{если } X > 2 \end{cases}$

<p>Вариант 19</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^3 + 5, & \text{если } X > 1 \\ 4 - x, & \text{если } X \leq 1 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x - 1 , & \text{если } X < 0 \\ 2x, & \text{если } 0 \leq x \leq 2 \\ x + 6, & \text{если } X > 2 \end{cases}$	<p>Вариант 20</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 1 - x^2, & \text{если } X < 0 \\ 3x, & \text{если } X \geq 0 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 3 - x, & \text{если } X < -3 \\ 2 + x^2, & \text{если } -3 \leq x \leq 2 \\ 1 - x , & \text{если } X > 2 \end{cases}$	<p>Вариант 21</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^3, & \text{если } X > 2 \\ 4 - 3x, & \text{если } X \leq 2 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 3x + 1, & \text{если } X < 3 \\ 4x, & \text{если } 3 \leq x \leq 5 \\ 2 - x, & \text{если } X > 5 \end{cases}$
<p>Вариант 22</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^3 - 2, & \text{если } X \leq 3 \\ 4 - x, & \text{если } X > 3 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x + 5, & \text{если } X < -4 \\ 3x^2, & \text{если } -4 \leq x \leq 2 \\ 2x - 1, & \text{если } X > 2 \end{cases}$	<p>Вариант 23</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 3x^2, & \text{если } X > -1 \\ 6 - x, & \text{если } X \leq -1 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 6x, & \text{если } X < 2 \\ 2 - x , & \text{если } 2 \leq x \leq 6 \\ 5 - x, & \text{если } X > 6 \end{cases}$	<p>Вариант 24</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x - 3, & \text{если } X < 2 \\ 2x^2 - 1, & \text{если } X \geq 2 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x + 5, & \text{если } X < -4 \\ 7x - 1, & \text{если } -4 \leq x \leq 5 \\ 1 - x^2, & \text{если } X > 5 \end{cases}$
<p>Вариант 25</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 2x - x^2, & \text{если } X > -1 \\ 2x, & \text{если } X \leq -1 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} x^2 - 1, & \text{если } X < -3 \\ 3 - x, & \text{если } -3 \leq x \leq 0 \\ 4x, & \text{если } X > 0 \end{cases}$	<p>Вариант 26</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} x^2 - 1, & \text{если } X \leq -6 \\ 8 + x, & \text{если } X > -6 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 2 - x, & \text{если } X < -7 \\ 4x, & \text{если } -7 \leq x \leq 1 \\ 9 - x^2, & \text{если } X > 1 \end{cases}$	<p>Вариант 27</p> <p>1. Составить программу вычисления функции:</p> $y = \begin{cases} 1 + x^2, & \text{если } X > 5 \\ 2 - x, & \text{если } X \leq 5 \end{cases}$ <p>2. Составить программу вычисления функции:</p> $y = \begin{cases} 2x, & \text{если } X < -1 \\ \sin x - 1, & \text{если } -1 \leq x \leq 3 \\ x^2, & \text{если } X > 3 \end{cases}$

Смотрим видео:

https://www.youtube.com/watch?v=dcnCI_-Uq4E&list=PLA0M1Bcd0w8x9TltCzZDhw0SatK1d10yy&index=1

<https://www.youtube.com/watch?v=3UEoc5iWZUw&list=PLA0M1Bcd0w8x9TltCzZDhw0SatK1d10yy&index=2>

Тема: Логические операторы.

`||` записывается на «правде»,
`&&` записывается на «лжи».

Логические операторы могут применяться к значениям любого типа и возвращают также значения любого типа.

Логический оператор `||` (ИЛИ)

Оператор ИЛИ возвращает то значение, на котором остановились вычисления, причём, не преобразованное к логическому типу, т.е. оператор ИЛИ вычисляет ровно столько значений, сколько необходимо – до первого true.

Раздаточный материал №43

```
result = a || b;
```

```
alert( true || true ); // true
alert( false || true ); // true
alert( true || false ); // true
alert( false || false ); // false
```

Если значение не логического типа – то оно к нему приводится в целях вычислений.

Раздаточный материал №44

```
if (1 || 0) { // сработает как if( true || false ), число 1 будет воспринято как true, а 0 – как false
  alert( 'верно' );
}
```

Обычно оператор ИЛИ используется в if, чтобы проверить, выполняется ли хотя бы одно из условий.

Раздаточный материал №45

```
let hour = 12,
    isWeekend = true;

if (hour < 10 || hour > 18 || isWeekend) {
  alert( 'Офис до 10 или после 18 или в выходной закрыт' );
}
```

JavaScript вычисляет несколько ИЛИ слева направо. Если первый аргумент – true, то результат заведомо будет true (хотя бы одно из значений – true), и остальные значения игнорируются. Если все значения «ложные», то ИЛИ возвратит последнее из них

Раздаточный материал №46

```
let x;
true || (x = 1);
alert(x); // undefined, x не присвоен

let x;
```

```
false || (x = 1);  
alert(x); // 1
```

```
let undef; // переменная не присвоена, т.е. равна undefined  
let zero = 0;  
let emptyStr = "";  
let msg = "Привет!";  
let result = undef || zero || emptyStr || msg || 0;  
alert( result ); // выведет "Привет!" - первое значение, которое является true
```

```
alert( undefined || " || false || 0 ); // 0
```

Логический оператор && (И)

Раздаточный материал №47

```
result = a && b;  
  
alert( true && true ); // true  
alert( false && true ); // false  
alert( true && false ); // false  
alert( false && false ); // false
```

Раздаточный материал №48

```
let hour = 12,  
    minute = 30;  
  
if (hour == 12 && minute == 30) {  
    alert( 'Время 12:30' );  
}
```

В логическом И допустимы любые значения. Если левый аргумент – false, оператор И возвращает его и заканчивает вычисления. Иначе – вычисляет и возвращает правый аргумент. Можно передать и несколько значений подряд, при этом возвратится первое «ложное» (на котором остановились вычисления), а если его нет – то последнее. Оператор && вычисляет операнды слева направо до первого «ложного» и возвращает его, а если все истинные – то последнее значение. Приоритет оператора И больше, чем ИЛИ.

Раздаточный материал №49

```
if (1 && 0) { // вычислится как true && false  
    alert ( 'не сработает, т.к. условие ложно' );  
}
```

```
// Первый аргумент - true,  
// Поэтому возвращается второй аргумент  
alert( 1 && 0 ); // 0  
alert( 1 && 5 ); // 5
```

```
// Первый аргумент - false,
```

```
// Он и возвращается, а второй аргумент игнорируется
alert( null && 5 ); // null
alert( 0 && "не важно" ); // 0

alert( 1 && 2 && null && 3 ); // null

alert( 1 && 2 && 3 ); // 3

alert( 5 || 1 && 0 ); // 5
```

Логический оператор ! (НЕ) получает один аргумент. Сначала приводит аргумент к логическому типу true/false. Затем возвращает противоположное значение.

Раздаточный материал №50

```
let result = !value;

alert( !true ); // false
alert( !0 ); // true
```

С преподавателем *Раздаточный материал №51*

Что выведет код `alert(alert(1) || 2 || alert(3));`

Ответ: сначала 1, затем 2.

Решение

Вызов `alert` не возвращает значения, или, иначе говоря, возвращает `undefined`. Первый оператор ИЛИ `||` выполнит первый `alert(1)`, получит `undefined` и пойдёт дальше, ко второму операнду. Так как второй операнд 2 является истинным, то вычисления завершатся, результатом `undefined || 2` будет 2, которое будет выведено внешним `alert(....)`. Второй оператор `||` не будет выполнен, выполнение до `alert(3)` не дойдёт, поэтому 3 выведено не будет.

Что выведет код `alert(alert(1) && alert(2));`

Ответ: 1, а затем `undefined`. Вызов `alert` не возвращает значения, или, иначе говоря, возвращает `undefined`. Поэтому до правого `alert` дело не дойдёт, вычисления закончатся на левом.

Самостоятельно *Раздаточный материал №52*

Что выведет код `alert(null || 2 || undefined);`

Ответ: 2, это первое значение, которое в логическом контексте даст `true`.

Что выведет код `alert(1 && null && 2);`

Ответ: `null`, это первое ложное значение из списка.

Что выведет код `alert(null || 2 && 3 || 4);`

Ответ: 3. Приоритет оператора `&&` выше, чем `||`, поэтому он выполнится первым. Последовательность вычислений:

`null || 2 && 3 || 4`

`null || 3 || 4`

3

Напишите условие `if` для проверки того факта, что переменная `age` находится между 14 и 90 включительно.

решение

```
if (age >= 14 && age <= 90)
```

Напишите условие if для проверки того факта, что age НЕ находится между 14 и 90 включительно. Сделайте два варианта условия: первый с использованием оператора НЕ !, второй – без этого оператора.

решение

Первый вариант:

```
if (!(age >= 14 && age <= 90))
```

Второй вариант:

```
if (age < 14 || age > 90)
```

Какие из этих if верны, т.е. выполняются? Какие конкретно значения будут результатами выражений в условиях if(...)?

```
if (-1 || 0) alert( 'первое' );
```

```
if (-1 && 0) alert( 'второе' );
```

```
if (null || -1 && 1) alert( 'третье' );
```

решение

Ответ: первое и третье выполняются.

// Выполнится

// Результат -1 || 0 = -1, в логическом контексте true

```
if (-1 || 0) alert( 'первое' );
```

// Не выполнится

// -1 && 0 = 0, в логическом контексте false

```
if (-1 && 0) alert( 'второе' );
```

// Выполнится

// оператор && имеет больший приоритет, чем ||

// так что -1 && 1 выполнится раньше

// вычисления: null || -1 && 1 -> null || 1 -> 1

```
if (null || -1 && 1) alert( 'третье' );
```

Тема: Работа с формами

Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

Устанавливают форму на веб-странице теги `<form>` и `</form>`. Тег имеет следующие атрибуты:

accept-charset - устанавливает кодировку, в которой сервер может принимать и обрабатывать данные, например Windows-1251, UTF-8 и др.

action - адрес программы или документа (обработчик), который обрабатывает данные формы. В качестве обработчика может выступать CGI-программа или HTML-документ, который включает в себя серверные сценарии. После выполнения обработчиком действий по работе с данными формы он возвращает новый HTML-документ. Если атрибут `action` отсутствует, текущая страница перезагружается, возвращая все элементы формы к их значениям по умолчанию. В качестве значения принимается полный или относительный путь к серверному файлу (URL).

HTML.
Раздаточный
материал №53

```
<form action="handler.php">
  <p>...</p>
</form>
```

HTML.
Раздаточный
материал №54

```
<form action="handler.php" method="post">
```

name – уникальное имя формы, которое чаще всего используется для обращения к форме из скриптов.

Допускается внутрь контейнера `<form>` помещать другие теги, при этом сама форма никак не отображается на веб-странице, видны только ее элементы и результаты вложенных тегов.

Элементы формы

На форме могут быть определены следующие элементы управления: кнопка, флажок, радиокнопка, меню, строка текста, пароль, скрытое поле, файл, объект. Основной атрибут тега `<input>`, определяющий вид элемента — `type`.

Строка текста

Позволяет вводить короткие фрагменты текста – имена, адреса и т.п. При его создании используется `name` для указания имени, которое поможет различить его в наборе данных на сервере. Наименование поля может состоять из букв, цифр и `_`.

Атрибуты:

`size` – ширина строки текста.

maxlength – ограничивает емкость строки текста.

value – начальный текст отображаемый в поле.

HTML.

Раздаточный
материал №55

```
<br> Ключевые слова: <input type="text" name="key" size="30"
maxlength="25">
```

type="password" – строка текста для хранения конфиденциальной информации.

Многострочный текст

<textarea> – создает область ввода многострочного текста.

name – имя области ввода.

rows, cols – ширина и высота области ввода

wrap= задает правила переноса целых слов текста:

soft – включает функцию переноса, но не воздействует на внешний вид данных, передаваемых на сервер.

hard – форматирование данных как на экране, так и во внутреннем буфере браузера.

HTML.

Раздаточный
материал №56

```
<textarea name="unit" rows="5" cols="45" wrap="hard">
</textarea>
```

Флажок

type="checkbox"

name – имя группы элементов-флажков.

value – служит для назначения имени каждому флажку в группе. Имя будет включено в итоговый набор данных формы, если пользователь выберет соответствующий флажок.

В значениях name и value используются только буквы, цифры и _.

checked – позволяет установить флажок предварительно, чтобы при воспроизведении страницы опция выбиралась автоматически.

HTML.

Раздаточный
материал №57

```
<br> Установите атрибуты файла
<br><input type="checkbox" name="flag" value="sys" checked>
системный
<br><input type="checkbox" name="flag" value="archive">архивный
<br><input type="checkbox" name="flag" value="read">только чтение
```

Переключатель

Используется при выборе только одной опции. Создание аналогично флажку. Type="radio".

Меню

name, value – аналогично флажку.

size – число элементов меню отображаемых одновременно (по умолчанию – один элемент).

<option> – определяет отдельный элемент меню.

selected – автоматический выбор опции.

```
<br> Какой цвет Вам нравится?
<br><select name="color" size="3">
  <option value="blue">Синий
  <option value="red" selected >Красный
  <option value="yellow">Желтый
</select>
<br><br><br><input type="submit" value="отправить">&nbsp;
<input type="reset" value="очистить">
```

Кнопка submit служит для завершения пользовательского сеанса заполнения формы и отправки данных на сервер.

Кнопка reset очищает все поля формы и возвращает их к исходному состоянию.

Value – определяет текст, отображаемый на поверхности кнопки (по умолчанию submit/reset).

Кроме кнопки Submit для отправки формы на сервер можно нажать клавишу Enter в пределах формы. Если кнопка Submit отсутствует в форме, клавиша Enter имитирует ее использование, но только в том случае, когда в форме имеется только один элемент <input>. Если таких элементов два и более, нажатие на <Enter> не вызовет никакого результата.

Второй способ создания кнопки основан на использовании тега <button>.

Button позволяет отображать разнообразные элементы в виде кнопок. Кнопки могут быть созданы с использованием всех возможностей, доступных в HTML и таблицах стилей (Поддерживается не всеми браузерами).

```
<button style="font-family:Arial; font-size:16pt; color:navy">добавьте
информацию<span style="font-style:italic; color:green> Пожалуйста!
</span>
</button>
```

Вариант 1	https://professorweb.ru/my/html/html5/level2/files/img46023.jpg
Вариант 2	https://cf.ppt-online.org/files/slide/v/vRAIxeaYzgFDVC2krBdPXQtZbW5McoLJ08In6q/slide-21.jpg
Вариант 3	https://lh5.googleusercontent.com/-wG_YHAibVZU/Ud696wJg0FI/AAAAAAAAACP4/ealzPTZRxE/w596-h642-no/4_3.png
Вариант 4	https://www.devlounge.net/wp-content/uploads/2011/02/24ways.jpg
Вариант 5	https://i.pinimg.com/originals/73/c6/0d/73c60def8c55043f9fd27b370530a9cf.jpg
Вариант 6	https://smashinghub.com/wp-content/uploads/2011/08/html5-forums-14.jpg
Вариант 7	https://soulcompas.com/wp-content/uploads/2020/06/html5-admin-template-free.jpg
Вариант 8	https://www.webasyst.ru/wa-data/public/updates/img/09/209/7355/7355.970.jpg
Вариант 9	https://www.zoho.com/creator/images/subpages/htmlforms/workshop_registration.gif
Вариант 10	https://bramus.github.io/ws1-sws-course-materials/assets/03/testform.jpg
Вариант 11	https://www.bestfree.ru/uslugi/constructors/SozdanieSaytaNaUcoz_3.png
Вариант 12	http://ip-whois.net/forma-obratnoj-svyazi/img/form1.jpg
Вариант 13	https://www.digiseller.ru/preview/125077/p1_30116215520413.JPG
Вариант 14	https://mob25.com/images/Perl/images/ris150_1.jpg
Вариант 15	https://i2.wp.com/ps.w.org/ultimate-form-builder-lite/assets/screenshot-1.png
Вариант 16	https://i.stack.imgur.com/px0jV.png

Тема: Функции в JavaScript.

При создании скрипта разумно в нем выделить логически независимые части – функции.

Раздаточный материал №61

```
function F(V)
{
S
}
```

где F – имя функции, по которому к ней можно обращаться,
V – список параметров, разделенных запятыми,
S – тело функции (операторы).

Описание функции не может быть вложено в описание другой функции. Начальные значения параметрам присваиваются при обращении к функции. Если описание функции имеет вид

Раздаточный материал №62

Function F(V1,V2,...,Vn) {S},
то вызов функции должен иметь вид F(e1,e2,...,en),
где e1,e2,...,en - выражения, задающие фактические значения параметров.

Параметры V1,V2,...,Vn – формальные параметры, которые получают смысл только после задания в функции фактических параметров e1,e2,...,en, с которыми функция затем и работает. Если в функции параметры отсутствуют, то описание функции следующее:

Раздаточный материал №63

```
Function F()
{
S
}
```

где F – имя функции, по которому к ней можно обращаться,
S – тело функции (операторы).

Вызов функции:

1. Указав ее имя в качестве оператора JavaScript.
2. Указав ее имя в HTML-ссылке
` ... `
3. В адресной строке
`javascript:hello()`
4. В форме при помощи атрибута action

Раздаточный материал №64

```
<html>
<head>
<script>
function hello() {
```

```
document.write("Привет и добро пожаловать");
}
</script>
</head>
<body>
<form action="javascript:hello()">
<input type="submit" value="Отобразить приветствие">
</form>
</body>
</html>
```

Решение задач. *Раздаточный материал №65*

1. Переработать функцию из раздаточного материала № 64 так, чтобы она сначала спрашивала имя пользователя, а затем выводила сообщение "Привет *имя_пользователя* и добро пожаловать".

2. Создать скрипт, в котором функция спросит имя пользователя и поприветствует его, а если пользователь не ввел имя, т.е. нажал «Отмена», то выдаст сообщение об этом.

```
<html>
<head>
<script>
function hello() {
user_name = prompt ("Введи имя", "Здесь");
if (user_name == null) {
    alert("Вы не ввели имя!");
}
else {
    document.write("Привет "+user_name+" и добро пожаловать");
}
}
</script>
</head>
<body>
<form action="javascript:hello()">
<input type="submit" value="Отобразить приветствие">
</form>
</body>
</html>
```

События и обработчики событий.

Кроме объектов и методов в JS имеются события и обработчики событий. Но они «встроены» HTML-код, а не существуют самостоятельно, как скрипты, т.е. не требуют тегов `<script>` и `</script>`. Это область взаимодействия страницы и пользователя.

Обработчики событий представляют собой небольшие подпрограммы, связывающие действия пользователей со сценариями, которые необходимо выполнить в ответ на эти действия. Обработчики событий можно помещать в HTML-теги и в атрибуты.

Событие **onMouseOver** срабатывает при наведении мыши на объект. Например, на гипертекстовую ссылку: используется внутри ее, при этом формат ссылки остается без изменений; обработчик ставится сразу же после адреса URL. Событие приводится в действие, как только будет распознано браузером.

Событие **onMouseOut** срабатывает после ухода мыши с объекта.

JavaScript. Раздаточный материал № 66

```
<a href="les10.htm" onMouseOver="alert('Уходи в сторону!')" onMouseOut="alert('Так-то лучше, спасибо')"> Наведите курсор на эту ссылку и уведите обратно</a>
```

Событие **onClick** срабатывает при однократном нажатии мыши.

JavaScript. Раздаточный материал № 67

```
<a href="http://www.jsp.newmail.ru" onClick="alert('Уже уходите!');"> Жмите сюда</a>
```

```
<input type="button" onclick="this.style.display='none'" value="Нажми, чтобы спрятать"/>
```

Событие **ondblClick** срабатывает при двойном нажатии мыши.

Событие **onMouseDown** срабатывает когда нажимается кнопка мыши.

Событие **onMouseUp** срабатывает когда кнопка мыши отпускается.

Событие **onFocus** срабатывает когда пользователь «фокусируется» на элементе страницы. Эта команда может использоваться для форм: флажков и текстовых полей.

JavaScript. Раздаточный материал № 68

```
<form>
<input type="text" size="30" onFocus="alert ('Вы остановились на элементе')">
</form>
```

Событие **onBlur** срабатывает при потере «фокуса», т.е. когда пользователь переходит к другому элементу

JavaScript. Раздаточный материал № 69

```
<form>
<input type="text" size="45" value="Впишите свое имя и щелкните по другой строке"
onBlur="alert('Вы изменили ответ — уверены, что он правильный?');">
</form>
```

Событие **onChange** срабатывает при изменении объекта. Обработчик события `onChange` полезен для проверки правильности заполнения форм, поскольку это событие наступает при каждом изменении элемента формы.

Например, если в форме присутствует текстовое поле, запрашивающее у пользователя целочисленное значение от 0 до 9, можно создать функцию, которая будет проверять правильность значения, введенного в текстовое поле.

JavaScript.Раздаточный материал № 70

```
<html>
<head>
<script>
function intCheck() {
val=document.form1.text1.value;
if (val<0||val>9)
alert("Это значение недопустимо. Введите целое число в пределах от 0 до 9");
}
</script>
</head>
<body>
<form name= "form1">
<input type="text" name="text1" onChange="intCheck()">
</form>
</body>
</html>
```

JavaScript.Раздаточный материал № 71

```
<form>
<input TYPE="text" size="45" value="Измените текст и щелкните по другой строке"
onChange="alert('Текст был изменен')">
</form>
```

Событие **onSelect** срабатывает при выделении объекта.

Событие **onSubmit** (отослать, отправить) срабатывает при нажатии кнопки Submit.

JavaScript.Раздаточный материал № 72

```
<form>
<input TYPE="submit" onSubmit="parent.location='thanksalot.html'">
</form>
```

parent.location – это стандартная схема ссылки на другую страницу.

parent – свойство окна браузера.

location – объект, который появиться в этом окне

Взаимодействие форм и JavaScript.

Часто взаимодействие формы и JavaScript используется для проверки ввода данных. При организации взаимодействия форм и JavaScript необходимо следить за иерархией объектов:

JavaScript. Раздаточный материал № 73

document.myform.fname.value

↑ ↑ ↑ ↑

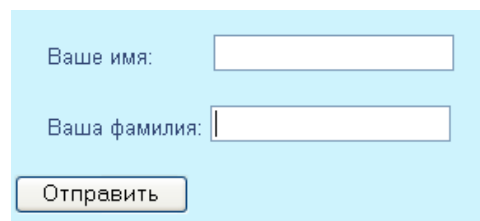
объект объект объект свойство

документ форма поле формы

Примеры взаимодействия форм и JavaScript

JavaScript. Раздаточный материал № 74

```
<html>
<head>
<script>
function doit()
{
    alert("document.myform.fname.value — это " + document.myform.fname.value)
    var greeting="Привет, "
    alert(greeting + document.myform.fname.value + " " + document.myform.lname.value)
    alert("Количество букв в имени " + document.myform.fname.value.length)
}
</script>
</head>
<body>
<form name="myform">
    Ваше имя:
        <input type="text" name="fname"><p>
    Ваша фамилия:
        <input type="text" name="lname"><p>
        <input type="button" value="отправить" onClick="doit()">
</form>
</body>
</html>
```



Ваше имя:

Ваша фамилия:

JavaScript. Раздаточный материал № 75

```
<html>
<head>
<script>
function newcolor(color)
{
    alert("Вы выбрали " + color);
    document.bgColor=color
}
</script>
</head>
```

```

<body>
<p>Выбрать цвет фона</p>
<form>
<input type="button" value="голубой" onClick="newcolor('lightblue')">
<input type="button" value="розовый" onClick="newcolor('pink')">
<input type="button" value="вернуть" onClick="newcolor('white')">
</form>
</body>
</html>

```

Раздаточный материал № 76

Задание 1.

Составьте документ HTML с формой aform. В ней должно быть два текстовых поля, одно для геометрической фигуры, другое для цвета, и кнопка. Напишите функцию с переменной, которая содержит слова «Мне нравится ». Когда пользователь нажмет на кнопку, должно всплывать окно со следующей надписью:

1. Мне нравится *геометрическая фигура такого-то цвета*. (по результатам тех данных, которые пользователь вводит в форму)
2. Покажите длину (length) «фигуры».

Задание 2.

Создать форму, которая будет взаимодействовать с пользователем. Форма должна иметь три элемента:

- поле ввода с просьбой ввести имя;
- два поля для флажков с вопросом о том, что больше нравится пользователю, мороженое или шоколад;
- кнопку отправки данных.

С каждым элементом должно произойти следующее:

- При вводе имени должно появиться окно предупреждения: «Внимательно вводите свое имя».
- Два поля с флажками должны выдавать окно со словами: «Вы выбрали...» и выбор пользователя.
- При нажатии на кнопку должно появиться окно предупреждения, благодарящее пользователя за участие в опросе.

Решение:

```

<form>
Name: <input type="text" size="30" onClick="alert('Внимательно вводите свое имя');"><br>
Что вам больше нравится: <br><input type="checkbox" onClick="alert('Вы выбрали Шоколад');">
Шоколад<br>
<input type="checkbox" onClick="alert('Вы выбрали Мороженое');"> Мороженое<br>
<input type="submit" onClick="alert('Спасибо за участие в опросе');">
</form>

```

Оператор switch

Используется в тех случаях, когда при решении задачи требуется выбрать один вариант действия из нескольких возможных. Синтаксис оператора:

Раздаточный материал № 77

```
switch (n)
{
  case L1: операторы; break;
  case L2: операторы; break;
  ...
  case Ln: операторы; break;
  default: операторы
}
```

Оператор break обеспечивает завершения работы оператора switch после выполнения очередного варианта.

Оператор break может отсутствовать. В этом случае операторы будут выполняться от L1 до конца оператора switch.

Раздаточный материал № 78

```
<!DOCTYPE html>
<script>

let a = Number(prompt('Введи число'));

switch (a) {
  case 3:
    alert( 'Маловато' );
    break;
  case 4:
    alert( 'В точку!' );
    break;
  case 5:
    alert( 'Перебор' );
    break;
  default:
    alert( "Нет таких значений" );
}
</script>
```

Раздаточный материал № 79

Перепишите код с использованием одной конструкции switch:

```
const number = +prompt('Введите число между 0 и 3, ');

if (number === 0) {
  alert('Вы ввели число 0');
}
```

```

if (number === 1) {
    alert('Вы ввели число 1');
}

if (number === 2 || number === 3) {
    alert('Вы ввели число 2, а может и 3');
}

```

Решение

```

const number = +prompt('Введите число между 0 и 3, ');

switch (number) {
    case 0:
        alert('Вы ввели число 0');
        break;

    case 1:
        alert('Вы ввели число 1');
        break;

    case 2:
    case 3:
        alert('Вы ввели число 2, а может и 3');
        break;
}

```

КТ № 4 (1 часть)

1. При открытии вклада в банке установлены следующие годовые проценты: при вкладе до 50000р. процент составит 4%; при вкладе от 50000р. до 100000р. процент составит 5%; при вкладе от 100000р. до 150000р. скидка составит 6%; при вкладе от 150000 р. до 200000 р. процент составит 7%. Составить программу, определяющую размер процентной ставки в зависимости от вносимой суммы.
2. Размер скидки на продукты определен следующим образом: при покупке до 500 р. скидка составит 2%; при покупке от 500 р. до 1000 р. скидка составит 3%; при покупке от 1000 р. до 1500 р. скидка составит 4%; при покупке от 1500 р. до 2000 р. скидка составит 5%. Составить программу определяющую размер скидки в зависимости от потраченной суммы.
3. Спектр видимого излучения представлен в таблице. Составить программу, определяющую название цвета в зависимости от введенной длины волны.

Цвет	Диапазон длин волн, нм
Фиолетовый	≤450
Синий	450—480
Сине-зелёный	480—510
Зелёный	510—550
Жёлто-зелёный	550—570
Жёлтый	570—590
Оранжевый	590—630
Красный	≥630

4. Мастям игральных карт присвоены порядковые номера: 1– пики, 2 – трефы, 3 – бубны, 4 – червы. Достоинству карт, старших десятки, присвоены номера: 11 – валет,

- 12 – дама, 13 – король, 14 – туз. Дано трехзначное число, в котором первая цифра указывает на масть, а вторые две на достоинство карты. Вывести соответствующее название карты вида «дама червей», «туз треф» и т.п.
5. Смоделировать простейший калькулятор, умеющий выполнять 4 основные арифметические операции
 6. Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы (целое число в диапазоне 1-5) и масса тела в этих единицах (вещественное число). Найти массу тела в килограммах.
 7. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины (целое число в диапазоне 1-5) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.
 8. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия N (целое число в диапазоне 1-4) и числа A и B (B не равно 0). Выполнить над числами указанное действие и вывести результат.
 9. Дан номер месяца — целое число в диапазоне 1-12 (1 — январь, 2 — февраль и т. д.). Определить количество дней в этом месяце для невисокосного года.
 10. Дано целое число, лежащее в диапазоне 1-999. Вывести его строку- описание вида «четное двузначное число», «нечетное трехзначное число» и т. д.
 11. Дано целое число. Вывести его строку-описание вида «отрицательное четное число», «нулевое число», «положительное нечетное число» и т. д.
 12. Составить программу, которая в зависимости от порядкового номера дня недели (1, 2, ..., 7) выводит на экран его название (понедельник, вторник, ..., воскресенье).
 13. Составить программу, которая в зависимости от порядкового номера дня месяца (1, 2, ..., 12) выводит на экран его название (январь, февраль, ..., декабрь).
 14. Составить программу, которая в зависимости от порядкового номера месяца (1, 2, ..., 12) выводит на экран время года, к которому относится этот месяц.
 15. При открытии вклада в банке установлены следующие годовые проценты: при вкладе до 50000р. процент составит 4%; при вкладе от 50000р. до 100000р. процент составит 5%; при вкладе от 100000р. до 150000р. скидка составит 6%; при вкладе от 150000 р. до 200000 р. процент составит 7%. Составить программу, определяющую размер процентной ставки в зависимости от вносимой суммы.
 16. Размер скидки на продукты определен следующим образом: при покупке до 500 р. скидка составит 2%; при покупке от 500 р. до 1000 р. скидка составит 3%; при покупке от 1000 р. до 1500 р. скидка составит 4%; при покупке от 1500 р. до 2000 р. скидка составит 5%. Составить программу определяющую размер скидки в зависимости от потраченной суммы.
 17. Спектр видимого излучения представлен в таблице. Составить программу, определяющую название цвета в зависимости от введенной длины волны.

Цвет	Диапазон длин волн, нм
Фиолетовый	≤450
Синий	450—480
Сине-зелёный	480—510
Зелёный	510—550
Жёлто-зелёный	550—570
Жёлтый	570—590
Оранжевый	590—630
Красный	≥630

18. Мастям игровых карт присвоены порядковые номера: 1— пики, 2 — трефы, 3 — бубны, 4 — червы. Достоинству карт, старших десятки, присвоены номера: 11 — валет, 12 — дама, 13 — король, 14 — туз. Дано трехзначное число, в котором первая цифра

указывает на масть, а вторые две на достоинство карты. Вывести соответствующее название карты вида «дама червей», «туз треф» и т.п.

19. Смоделировать простейший калькулятор, умеющий выполнять 4 основные арифметические операции
20. Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы (целое число в диапазоне 1-5) и масса тела в этих единицах (вещественное число). Найти массу тела в килограммах.
21. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины (целое число в диапазоне 1-5) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.
22. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия N (целое число в диапазоне 1-4) и числа A и B (B не равно 0). Выполнить над числами указанное действие и вывести результат.
23. Дан номер месяца — целое число в диапазоне 1-12 (1 — январь, 2 — февраль и т. д.). Определить количество дней в этом месяце для невисокосного года.
24. Дано целое число, лежащее в диапазоне 1-999. Вывести его строку- описание вида «четное двузначное число», «нечетное трехзначное число» и т. д.

Циклы в JavaScript.

JavaScript поддерживает три вида циклов: for, while, do..while.

Раздаточный материал № 80

```
For (начальное значение; конечное значение; шаг)
{
  Тело цикла
}
```

JavaScript. Раздаточный материал № 81

```
<script>
  function countRabbits() {
    for(let i=1; i<=3; i++) {
      alert("Кролик номер " + i);
    }
  }
</script>
</head>
<body>
  <input type="button" onclick="countRabbits()" value="Считать кроликов!"/>
</body>
</html>
```

Пропуск частей for

Любая часть for может быть пропущена. Например, можно убрать начало.

Раздаточный материал № 82

```
let i = 0;

for (; i < 3; i++) {
  alert(i); // 0, 1, 2
}
```

Можно убрать и шаг:

JavaScript. Раздаточный материал № 83

```
let i = 0;
for (; i < 3;) {
  alert(i);
  // цикл превратился в аналог while (i<3)
}
```

А можно и вообще убрать всё, получив бесконечный цикл:

Раздаточный материал № 84

```
for (;;) {  
    // будет выполняться вечно  
}
```

При этом сами точки с запятой ; обязательно должны присутствовать, иначе будет ошибка синтаксиса.

Цикл **while** имеет вид:

Раздаточный материал № 85

```
while (условие) {  
    // код, тело цикла  
}
```

Пока условие верно – выполняется код из тела цикла. Условие в скобках интерпретируется как логическое значение, поэтому вместо while (i!=0) обычно пишут while (i)

Раздаточный материал № 86

```
let i = 0;  
while (i < 3) {  
    alert( i );  
    i++;  
}
```

```
let i = 3;  
while (i) { // при i, равном 0, значение в скобках будет false и цикл остановится  
    alert( i );  
    i--;  
}
```

Цикл **do...while** имеет вид:

Цикл сначала выполняет тело, а затем проверяет условие.

Раздаточный материал № 87

```
do {  
    // тело цикла  
} while (условие);
```

Раздаточный материал № 88

```
let i = 0;  
do {  
    alert( i );  
    i++;  
} while (i < 3);
```

В цикле также можно определить переменную:

JavaScript. Раздаточный материал № 89

```
for (let i = 0; i < 3; i++) {  
  alert(i); // 0, 1, 2  
}
```

Эта переменная будет видна и за границами цикла, в частности, после окончания цикла `i` станет равно 3.

Прерывание цикла.

Оператор **break** автоматически завершает цикл и выполняет первый оператор, следующий за оператором цикла. При этом все переменные остаются в том состоянии, в котором они были при завершении цикла.

Раздаточный материал № 90

```
let sum = 0;  
While (true) {  
  let value = +prompt("Введите число", "");  
  if (!value) break; // (*)  
  sum += value;  
}  
alert( 'Сумма: ' + sum );
```

Директива **continue** прекращает выполнение текущей итерации цикла, когда понятно, что на текущей итерации цикла больше выполнять ничего не нужно.

JavaScript. Раздаточный материал № 91

```
for (let i = 0; i < 10; i++) {  
  if (i % 2 == 0) continue;  
  alert(i);  
}
```

Для чётных `i` срабатывает `continue`, выполнение тела прекращается и управление передаётся на следующий проход `for`.

Нельзя использовать `break/continue` справа от оператора „?“

Метки для `break/continue` позволяют выйти одновременно из нескольких уровней цикла.

Раздаточный материал № 92

```
outer: for (let i = 0; i < 3; i++) {  
  for (var j = 0; j < 3; j++) {  
    let input = prompt("Значение в координатах 'i+', 'j', ");  
    // если отмена ввода или пустая строка -  
    // завершить оба цикла  
    if (!input) break outer; // (*)  
  }  
}  
alert('Готово!');
```

Будет разорван самый внешний цикл и управление перейдёт на `alert`

Метка имеет вид "имя:" (имя должно быть уникальным) и ставится перед циклом или на отдельной строке. Вызов break outer ищет ближайший внешний цикл с такой меткой и переходит в его конец.

Директива continue также может быть использована с меткой, в этом случае управление перейдет на следующую итерацию цикла с меткой.

Метки не позволяют перейти в произвольное место кода, в JavaScript нет такой возможности.

<https://www.youtube.com/watch?v=oc6HYRgY5uw&list=PLA0M1Bcd0w8x9TltCzZDhw0SatK1d10yy&index=7>

КТ № 4 (2 часть)

<p>Вариант 1</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{b-1}{i+2}$	<p>Вариант 2</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{1+a^2}{\sqrt{i(i+1)}}$	<p>Вариант 3</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{3i-c}{1+\sqrt{i}}$
<p>Вариант 4</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{i^2+2a}{i}$	<p>Вариант 5</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \sin(i^2+3b)$	<p>Вариант 6</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{4xi^2}{1+i}$
<p>Вариант 7</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{a^2+3i}{3+i}$	<p>Вариант 8</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n 3b + \frac{2b}{i^2}$	<p>Вариант 9</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{x\sqrt{1+3i}}{2+i}$
<p>Вариант 10</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{4xi^2+1}{i-1}$	<p>Вариант 11</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{\sin i^2+2ai}{3i+1}$	<p>Вариант 12</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n (2i + \sqrt{3i+b^2})$
<p>Вариант 13</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n (\frac{1}{2}i^2x - 4x)$	<p>Вариант 14</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{\sqrt{3a^2+i^3}}{2i^2+3}$	<p>Вариант 15</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{2x+i^2}{i^2+1}$
<p>Вариант 16</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n (i^2 - 5bi^2)$	<p>Вариант 17</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{1+\sqrt{i+x^2}}{4i^2}$	<p>Вариант 18</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{2a^2+i^3}{i^2+2i}$
<p>Вариант 19</p> <p>1. Составить программу вычисления функции</p>	<p>Вариант 20</p> <p>1. Составить программу вычисления функции</p>	<p>Вариант 21</p> <p>1. Составить программу вычисления функции</p>

$y = \sum_{i=1}^n \frac{xi^3 + 3x}{2 + i}$	$y = \sum_{i=1}^n \frac{i^2 + 2ai}{\sqrt{3i^2 + i}}$	$y = \sum_{i=1}^n (i^2 - \frac{2b^2}{3i})$
<p>Вариант 22</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n (x^2i + 2x^3)$	<p>Вариант 23</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{3i + 2a}{a^2 + 2i}$	<p>Вариант 24</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{4bi^2}{1 + i \cos i}$
<p>Вариант 25</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{x^2 + 3i^2}{3x^2 + i}$	<p>Вариант 26</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n 2a^2 + \frac{2 \sin i}{i^2}$	<p>Вариант 27</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{b\sqrt{1 + 3i}}{2 + i}$
<p>Вариант 28</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{4xi^2 + 1}{i + x}$	<p>Вариант 29</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{\sin i^2 + 2xi}{3i^2 + 1}$	<p>Вариант 30</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n (2ia + \sqrt{3i + a^2})$
<p>Вариант 31</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{\cos 2x + 3i1}{i^2 + 2}$	<p>Вариант 32</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n \frac{i^2 + 2xi}{3i + \sin^2 x}$	<p>Вариант 33</p> <p>1. Составить программу вычисления функции</p> $y = \sum_{i=1}^n (2\cos i^2 + \sqrt{3 + 2a^2})$

Одномерные массивы с числовыми индексами

Массив представляет собой набор элементов, доступ к которым осуществляется по индексу. Один массив может включать несколько различных типов переменных. Все элементы массива пронумерованы, начиная с нуля. Для получения значения элемента массива необходимо задать имя массива и в квадратных скобках – порядковый номер элемента.

Раздаточный материал № 93

```
let land = [ ] //пустой массив

let land = [“Планета”, “24 часа”] // массив land с двумя элементами

alert(land[1]) // результат – 24 часа

land[3] = 6378 //добавлен третий элемент – радиус, теперь
[“Планета”, “24 часа”, 6378]

land[3] = 6371 //замена значения третьего элемента, теперь
[“Планета”, “24 часа”, 6371]

alert(land) // вывести массив целиком, через запятую
Планета, 24 часа, 6371
```

В массиве могут храниться элементы любого типа.

Раздаточный материал № 94

```
// разные типы значений
let arr = [ 'Яблоко', { name: 'Джон' }, true, function() { alert('привет'); } ];

// получить элемент с индексом 1 (объект) и затем показать его свойство
alert( arr[1].name ); // Джон

// получить элемент с индексом 3 (функция) и выполнить её
arr[3](); // привет
```

Методы объекта Array

Раздаточный материал № 95

sort	сортирует элементы массива
reverse	переставляет элементы в обратном порядке
push	добавляет один или несколько элементов в конец массива
pop	удаляет последний элемент из массива
unshift	добавляет один или несколько элементов в начало массива
shift	удаляет первый элемент массива
join	соединяет элементы массива в одну строку
concat	объединяет два массива в один
split	обеспечивает разделение строк (подробнее см. РМ № 82)
slice	копирует часть массива от начального до конечного элемента. Конечный элемент не копируется. Исходный массив не изменяется.


```
<script>
let land = ["Планета", "24 часа", 6371];
alert(land.pop() ); // удален 6371
alert(land); // Планета, 24 часа
</script>
```

```
<html>
  <body>
    <script>
      //1
      let names = 'Иванов, Петров, Сидоров';
      let fio = names.split(', ');
      console.log('Первый способ');
      for (let f of fio) {
        console.log('Привет, товарищ ' + f);
      }
    </script>

    <script>
      //2
      let names1 = 'Иванов, Петров, Сидоров';
      let fio1 = names.split(', ');
      console.log('Второй способ');
      for (let i = 0; i < fio1.length; i++) {
        console.log('Привет, товарищ ' + fio1[i]);
      }
    </script>

  </body>
</html>
```

```
<script>
let str = "слово";
alert(str.split(""));
</script>
```

```
<script>
let arr = ['Иванов', 'Петров', 'Сидоров'];
let str = arr.join(';');
alert(str);
</script>
```

```
<script>
let arr = ["воскресенье", "понедельник", "вторник", "среда", "четверг", "пятница",
"суббота"];
let arr2 = arr.slice(2, 5);
alert( arr2 ); //вторник, среда, четверг
</script>
```

Свойство length позволяет определить число элементов в массиве.

Найти сумму пяти элементов одномерного массива. Результат вывести в консоль.

```
<html>
  <body>
    <script>
      let number = [1, 2, 3, 4, 5];
      let summ = 0;
      let i;

      for (i = 0; i < 5; i++) {
        summ += number[i];
      }
      console.log('Сумма элементов массива = ' + summ);
    </script>
  </body>
</html>
```

КТ № 4 (3 часть)

Вариант 1

Задание 1

Создайте массив styles с элементами «Джаз» и «Блюз».
Добавьте «Рок-н-ролл» в конец.
Удалите первый элемент массива.
Вставьте Рэп и Регги в начало массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти разницу элементов массива. Результат вывести в консоль.

Вариант 2

Задание 1

Создайте массив Magnit с элементами «Мясо», «Молоко».
Вставьте «Соль» и «Сахар» в начало массива.
Добавьте «Сыр» в конец.
Удалите первый элемент массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти произведение элементов массива. Результат вывести в консоль.

Вариант 3

Задание 1

Создайте массив Voyage с элементами «Карелия», «Абхазия».
Вставьте «Золотое кольцо» и «Волга» в начало массива.
Удалите первый элемент массива.
Добавьте «Мексика» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти сумму квадратов элементов массива. Результат вывести в консоль.

Вариант 4

Задание 1

Создайте массив Books с элементами «Лермонтов», «Достоевский».
Удалите первый элемент массива.
Вставьте «Тютчев» и «Пушкин» в начало массива.
Добавьте «Толстой» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти удвоенную сумму элементов массива.
Результат вывести в консоль.

Вариант 5

Задание 1

Создайте массив styles с элементами «Джаз» и «Блюз».
Добавьте «Рок-н-ролл» в конец.
Удалите первый элемент массива.
Вставьте Рэп и Регги в начало массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти сумму кубов элементов массива.
Результат вывести в консоль.

Вариант 6

Задание 1

Создайте массив Magnit с элементами «Мясо», «Молоко».
Вставьте «Соль» и «Сахар» в начало массива.
Добавьте «Сыр» в конец.
Удалите первый элемент массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, увеличить каждый элемент на 7. Результат вывести в консоль.

Вариант 7

Задание 1

Создайте массив Voyage с элементами «Карелия», «Абхазия».
Вставьте «Золотое кольцо» и «Волга» в начало массива.
Удалите первый элемент массива.
Добавьте «Мексика» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, к каждому элементу прибавить его квадрат.
Результат вывести в консоль.

Вариант 8

Задание 1

Создайте массив Books с элементами «Лермонтов», «Достоевский».
Удалите первый элемент массива.
Вставьте «Тютчев» и «Пушкин» в начало массива.
Добавьте «Толстой» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, от каждого элемента вычесть значение его куба.
Результат вывести в консоль.

Вариант 9

Задание 1

Создайте массив `styles` с элементами «Джаз» и «Блюз».
Добавьте «Рок-н-ролл» в конец.
Удалите первый элемент массива.
Вставьте Рэп и Регги в начало массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти разность кубов элементов массива.
Результат вывести в консоль.

Вариант 10

Задание 1

Создайте массив `Magnit` с элементами «Мясо», «Молоко».
Вставьте «Соль» и «Сахар» в начало массива.
Добавьте «Сыр» в конец.
Удалите первый элемент массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, от каждого элемента вычесть значение его квадрата. Результат вывести в консоль.

Вариант 11

Задание 1

Создайте массив `Voyage` с элементами «Карелия», «Абхазия».
Вставьте «Золотое кольцо» и «Волга» в начало массива.
Удалите первый элемент массива.
Добавьте «Мексика» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти разницу элементов массива. Результат вывести в консоль.

Вариант 12

Задание 1

Создайте массив `Books` с элементами «Лермонтов», «Достоевский».
Удалите первый элемент массива.
Вставьте «Тютчев» и «Пушкин» в начало массива.
Добавьте «Толстой» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти произведение элементов массива.
Результат вывести в консоль.

Вариант 13

Задание 1

Создайте массив `styles` с элементами «Джаз» и «Блюз».
Добавьте «Рок-н-ролл» в конец.
Удалите первый элемент массива.
Вставьте Рэп и Регги в начало массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти сумму квадратов элементов массива.
Результат вывести в консоль.

Вариант 14

Задание 1

Создайте массив Magnit с элементами «Мясо», «Молоко».
Вставьте «Соль» и «Сахар» в начало массива.
Добавьте «Сыр» в конец.
Удалите первый элемент массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти удвоенную сумму элементов массива.
Результат вывести в консоль.

Вариант 15

Задание 1

Создайте массив Voyage с элементами «Карелия», «Абхазия».
Вставьте «Золотое кольцо» и «Волга» в начало массива.
Удалите первый элемент массива.
Добавьте «Мексика» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти сумму кубов элементов массива.
Результат вывести в консоль.

Вариант 16

Задание 1

Создайте массив Books с элементами «Лермонтов», «Достоевский».
Удалите первый элемент массива.
Вставьте «Тютчев» и «Пушкин» в начало массива.
Добавьте «Толстой» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, увеличить каждый элемент на 7. Результат вывести в консоль.

Вариант 17

Задание 1

Создайте массив styles с элементами «Джаз» и «Блюз».
Добавьте «Рок-н-ролл» в конец.
Удалите первый элемент массива.
Вставьте Рэп и Регги в начало массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, к каждому элементу прибавить его квадрат.
Результат вывести в консоль.

Вариант 18

Задание 1

Создайте массив Magnit с элементами «Мясо», «Молоко».
Вставьте «Соль» и «Сахар» в начало массива.
Добавьте «Сыр» в конец.
Удалите первый элемент массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, от каждого элемента вычесть значение его куба.
Результат вывести в консоль.

Вариант 19

Задание 1

Создайте массив Voyage с элементами «Карелия», «Абхазия».
Вставьте «Золотое кольцо» и «Волга» в начало массива.
Удалите первый элемент массива.
Добавьте «Мексика» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти разность кубов элементов массива.
Результат вывести в консоль.

Вариант 20

Задание 1

Создайте массив Books с элементами «Лермонтов», «Достоевский».
Удалите первый элемент массива.
Вставьте «Тютчев» и «Пушкин» в начало массива.
Добавьте «Толстой» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов, от каждого элемента вычесть значение его квадрата. Результат вывести в консоль.

Вариант 21

Задание 1

Создайте массив styles с элементами «Джаз» и «Блюз».
Добавьте «Рок-н-ролл» в конец.
Удалите первый элемент массива.
Вставьте Рэп и Регги в начало массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти разницу элементов массива. Результат вывести в консоль.

Вариант 22

Задание 1

Создайте массив Magnit с элементами «Мясо», «Молоко».
Вставьте «Соль» и «Сахар» в начало массива.
Добавьте «Сыр» в конец.
Удалите первый элемент массива.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти произведение элементов массива.
Результат вывести в консоль.

Вариант 23

Задание 1

Создайте массив Voyage с элементами «Карелия», «Абхазия».
Вставьте «Золотое кольцо» и «Волга» в начало массива.
Удалите первый элемент массива.
Добавьте «Мексика» в конец.
Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти сумму квадратов элементов массива.
Результат вывести в консоль.

Вариант 24

Задание 1

Создайте массив Books с элементами «Лермонтов», «Достоевский».

Удалите первый элемент массива.

Вставьте «Тютчев» и «Пушкин» в начало массива.

Добавьте «Толстой» в конец.

Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти удвоенную сумму элементов массива.

Результат вывести в консоль.

Вариант 25

Задание 1

Создайте массив styles с элементами «Джаз» и «Блюз».

Добавьте «Рок-н-ролл» в конец.

Удалите первый элемент массива.

Вставьте Рэп и Регги в начало массива.

Результат вывести в консоль.

Задание 2

Создать одномерный массив на семь элементов. Найти сумму кубов элементов массива.

Результат вывести в консоль.

Строки

В JavaScript любые текстовые данные являются строками. Не существует отдельного типа «символ», который есть в ряде других языков.

Внутренний формат для строк — всегда UTF-16, вне зависимости от кодировки страницы.

В JavaScript есть разные типы кавычек. Строку можно создать с помощью одинарных (‘), двойных (“) либо обратных (`) кавычек:

Раздаточный материал № 97

```
let single = 'single-quoted';
let double = "double-quoted";
let backticks = `backticks`;
```

Одинарные и двойные кавычки работают, по сути, одинаково. Одним из преимуществ, обратных кавычек — они могут занимать более одной строки:

Раздаточный материал № 98*

```
let guestList = `Guests:
* John
* Pete
* Mary
`;
alert(guestList); // список гостей, состоящий из нескольких строк
```

Если попытаться использовать точно так же одинарные или двойные кавычки, то будет ошибка.

Свойство `length` содержит длину строки:

Раздаточный материал № 99*

```
<script>
    alert("Это строка".length) // выведет 10
</script>
```

Доступ к символам

Получить символ, который занимает позицию `pos`, можно с помощью квадратных скобок: `[pos]`. Первый символ занимает нулевую позицию:

Раздаточный материал № 100*

```
let str = `Hello`;
// получаем первый символ
alert( str[0] ); // H
```

Содержимое строки в JavaScript нельзя изменить. Нельзя взять символ посередине и заменить его. Как только строка создана — она такая навсегда.

Раздаточный материал № 101

```
let str = 'Hi';
str[0] = 'h'; // ошибка
alert( str[0] ); // не работает
```


Можно создать новую строку и записать её в ту же самую переменную вместо старой.

Раздаточный материал № 102*

```
<script>
let str = 'Hi';
str = 'h' + str[1]; // заменяем строку
alert( str ); // hi
</script>
```

Изменение регистра

Методы `toLowerCase()` и `toUpperCase()` меняют регистр символов:

Раздаточный материал № 103*

```
alert( 'Interface'.toUpperCase() ); // INTERFACE
alert( 'Interface'.toLowerCase() ); // interface
```

Если необходимо перевести в нижний регистр какой-то конкретный символ:

Раздаточный материал № 104*

```
alert( 'Interface'[0].toLowerCase() ); // 'i'
```

Поиск подстроки

Существует несколько способов поиска подстроки.

Метод — `str.indexOf(substr, pos)`. Он ищет подстроку `substr` в строке `str`, начиная с позиции `pos`, и возвращает позицию, на которой располагается совпадение, либо `-1` при отсутствии совпадений.

Раздаточный материал № 105*

```
let str = 'Widget with id';
alert( str.indexOf('Widget') ); // 0, потому что подстрока 'Widget' найдена в начале
alert( str.indexOf('widget') ); // -1, совпадений нет, поиск чувствителен к регистру
alert( str.indexOf("id") ); // 1, подстрока "id" найдена на позиции 1 (..idget with id)
```

Необязательный второй аргумент позволяет начать поиск с определённой позиции.

Раздаточный материал № 106*

Для того, чтобы найти следующее вхождение "id", начнём поиск с позиции 2:

```
let str = 'Widget with id';
alert( str.indexOf('id', 2) ) // 12
```

Метод `str.lastIndexOf(substr, position)` - ищет подстроку с конца строки к её началу. Он используется тогда, когда нужно получить самое последнее вхождение: перед концом строки или начинающееся до (включительно) определённой позиции.

Получение подстроки

В JavaScript есть 3 метода для получения подстроки: `substring`, `substr` и `slice`.

Метод `str.slice(start [, end])` - возвращает часть строки от `start` до (не включая) `end`.

Раздаточный материал № 107*

```
let str = "stringify";
alert( str.slice(0, 5) ); // 'strin', символы от 0 до 5 (не включая 5)
alert( str.slice(0, 1) ); // 's', от 0 до 1, не включая 1, т. е. только один символ на позиции 0
```

Если аргумент `end` отсутствует, `slice` возвращает символы до конца строки:

Раздаточный материал № 108*

```
let str = "stringify";
alert( str.slice(2) ); // ringify, с позиции 2 и до конца
```

Также для `start/end` можно задавать отрицательные значения. Это означает, что позиция определена как заданное количество символов с конца строки:

Раздаточный материал № 109*

```
let str = "stringify";
// начинаем с позиции 4 справа, а заканчиваем на позиции 1 справа
alert( str.slice(-4, -1) ); // gif
```

Метод `str.substring(start [, end])` - возвращает часть строки между `start` и `end`. Это — почти то же, что и `slice`, но можно задавать `start` больше `end`.

Раздаточный материал № 110*

```
let str = "stringify";
// для substring эти два примера — одинаковы
alert(str.substring(2, 6) ); // "ring"
alert(str.substring(6, 2) ); // "ring"

// ...но не для slice:
alert(str.slice(2, 6) ); // "ring" (то же самое)
alert(str.slice(6, 2) ); // "" (пустая строка)
```

Отрицательные значения `substring`, в отличие от `slice`, не поддерживает, они интерпретируются как 0.

Метод `str.substr(start [, length])` - возвращает часть строки от `start` длины `length`. В противоположность предыдущим методам, этот позволяет указать длину вместо конечной позиции:

Раздаточный материал № 111*

```
let str = "stringify";
alert(str.substr(2, 4) ); // ring, получаем 4 символа, начиная с позиции 2
```

Значение первого аргумента может быть отрицательным, тогда позиция определяется с конца:

Раздаточный материал № 112*

```
let str = "stringify";  
// gi, получаем 2 символа, начиная с позиции 4 с конца строки  
alert( str.substr(-4, 2) );
```

Раздаточный материал № 113*

Задания:

1. Выполнить все раздаточные материалы, помеченные (*).
2. Выполнить упражнения:
 - Создать строковую переменную, состоящую из нескольких строк (Films: The Terminator, Twilight, The Fast and the Furious, The Lord of the Rings).
 - Создать строковую переменную fraza = “Все произведения Достоевского актуальны в настоящее время”
 - Найти длину полученной строки fraza.
 - Последовательно вывести в консоль символы строковой переменной fraza, стоящие на позициях 1, 7, 14, 17, 21.
 - В строковой переменной fraza символы из верхнего регистра перевести в нижний.
 - В строковой переменной fraza символы из нижнего регистра перевести в верхний.
 - В строковой переменной fraza найти номер позиции слова «Достоевский» (str.indexOf(substr, pos)).
 - Из строковой переменной fraza последовательно вывести на экран (alert) все слова, при этом использовать методы substring, substr и slice.

3. Просмотр видео (2 шт.) по адресу:

<https://www.youtube.com/watch?v=Rvn5x8vGGGg&list=PLA0M1Bcd0w8x9TltCzZDhw0SatK1d10yy&index=17>
https://www.youtube.com/watch?v=-M_10IJV1wk&list=PLA0M1Bcd0w8x9TltCzZDhw0SatK1d10yy&index=14

Сравнение строк.

Строки сравниваются побуквенно. При этом сравниваются численные коды символов. Код у строчной буквы больше, чем у прописной. Для корректного сравнения символы должны быть в одинаковом регистре.

В частности, код у символа Б больше, чем у А, поэтому и результат сравнения такой.

Раздаточный материал № 114

```
alert( 'Б' > 'А' ); // true
```

Если строка состоит из нескольких букв, то сначала сравниваются первые буквы, потом вторые, и так далее, пока одна не будет больше другой.

Если первая буква первой строки больше – значит первая строка больше, независимо от остальных символов. Если одинаковы – сравнение идёт дальше. При этом любая буква больше отсутствия буквы.

Раздаточный материал № 115

```
alert( 'Привет' > 'Прив' ); // true, так как 'е' больше чем "ничего"
```

Что бы корректно сравнить строки необходимо использовать метод str.localeCompare(str2), который возвращает число, которое показывает, какая строка больше в соответствии с правилами языка:

больше в соответствии с правилами языка:

Отрицательное число, если str меньше str2.
Положительное число, если str больше str2.
0, если строки равны.

Раздаточный материал № 116

```
<script>  
    alert( 'Привет'.localeCompare( 'Прив' ) )  
  
</script>
```

Метод str.codePointAt(pos) возвращает код для символа, находящегося на позиции pos:

Раздаточный материал № 117*

```
// одна и та же буква в нижнем и верхнем регистре  
// будет иметь разные коды  
alert( "z".codePointAt(0) ); // 122  
alert( "Z".codePointAt(0) ); // 90
```

Метод String.fromCodePoint(code) отображает символ по его коду code.

Раздаточный материал № 118

```
alert( String.fromCodePoint(90) ); // Z
```

КТ № 5 (1 часть)

Задание 1

Средствами языка JavaScript составить слова из заданного слова:

Вариант 1.	Антресоль	нос, рот, сало, лень, налет
Вариант 2.	Антресоль	лен, тон, енот, лето, арест
Вариант 3.	Антресоль	лес, сто, есть, лось, лента
Вариант 4.	Антресоль	нет, лес, лань, ноль, лесть,
Вариант 5.	Антресоль	ось, оса, лена, нора, олень
Вариант 6.	Содержание	дар, даже, нора, адрес, держание
Вариант 7.	Содержание	дно, дерн, один, десна, дрожание
Вариант 8.	Содержание	еда, джин, одна, драже, оседание
Вариант 9.	Содержание	жар, дина, одна, народ, рождение
Вариант 10.	Содержание	жир, жанр, рано, недра, середина
Вариант 11.	Содержание	Ира, жена, реже, орден, сражение
Вариант 12.	Содержание	нос, Иран, роса, осина, осаждение
Вариант 13.	Содержание	ода, надо, сани, радио, средние
Вариант 14.	Содержание	она, неси, сено, ранее, седина
Вариант 15.	Содержание	они, ниже, сода, редис, сирена
Вариант 16.	Программирование	вон, вена, ангар, имена, аврора
Вариант 17.	Программирование	вор, вера, аниме, напев, ворона
Вариант 18.	Программирование	ген, вина, арена, напор, гарнир
Вариант 19.	Программирование	ива, воин, вагон, новое, многие
Вариант 20.	Программирование	маг, гимн, варан, номер, многое
Вариант 21.	Программирование	миг, гнев, верно, норма, мрамор
Вариант 22.	Программирование	мир, гном, ворон, овраг, нагрев
Вариант 23.	Программирование	мне, гора, гамма, опера, перрон
Вариант 24.	Программирование	мои, грим, грамм, опора, правое

Вариант 25.	Программирование	Пар, гром, грани, паром, пример
--------------------	------------------	---------------------------------

Задание 2

На основе примера из файла «Emodji.html» создать собственное изображение, используя метод `String.fromCodePoint(code)`.

Символы Эмодзи или любые другие:

<https://unicode-table.com/ru/> - таблица символов Unicode

Объект Math

(Ссылка на Булгаков

https://rostov.bulgakov.app/lessons/2439?blockId=content_anchor_0)

Объект Math включает свойства и методы для работы с математическими формулами, математическими постоянными и для выполнения вычислений. Объект Math не требуется специфицировать как подобъект объектов document или window, но его имя необходимо писать с прописной буквы.

abs()

Функция abs() возвращает абсолютное значение числа:

```
let x = -25;
console.log(Math.abs(x)); // 25
let y = 34;
console.log(Math.abs(y)); // 34
```

ceil()

Функция ceil() округляет число до следующего наибольшего целого числа:

```
let x = Math.ceil(9.2); // 10
let y = Math.ceil(-5.9); // -5
```

Выражение Math.ceil(9.2) возвращает число 10, так как число 10 следующее наибольшее целое число после 9.2. И также выражение Math.ceil(-5.9) возвращает -5, потому что число -5 следующее наибольшее целое после -5.9

floor()

Функция floor() округляет число до следующего наименьшего целого числа:

```
let x = Math.floor(9.2); // 9
let y = Math.floor(-5.9); // -6
```

round()

Функция round() округляет число до следующего наименьшего целого числа, если его десятичная часть меньше 0.5. Если же десятичная часть равна или больше 0.5, то округление идёт до ближайшего наибольшего целого числа:

```
let x = Math.round(5.5); // 6
let y = Math.round(5.4); // 5
let z = Math.round(-5.4); // -5
let n = Math.round(-5.5); // -5
let m = Math.round(-5.6); // -6
console.log(x);
console.log(y);
console.log(z);
console.log(n);
```

pow()

Функция pow() возвращает число в определенной степени. Например, возведём число 2 в степень 3:

```
let x = Math.pow(2, 3); // 8
```

sqrt()

Функция sqrt() возвращает квадратный корень числа:

```
let x = Math.sqrt(121); // 11
let y = Math.sqrt(9); // 3
let z = Math.sqrt(20); // 4.47213595499958
```

log()

Функция log() возвращает натуральный логарифм (по основанию e) числа:

```
let x = Math.log(1); // 0
let z = Math.log(10); // 2.302585092994046
```

Тригонометрические функции

Целый ряд функций представляют тригонометрические функции: sin() (вычисляет синус угла), cos() (вычисляет косинус угла), tan() (вычисляет тангенс угла).

```
let x = Math.sin(90); // 0.8939966636005579
let y = Math.cos(0); // 1
let z = Math.tan(45); // 1.6197751905438615
```

Функция asin() вычисляет арксинус числа, acos() - арккосинус, а atan() - арктангенс числа:

```
let x = Math.asin(0.9); // 1.1197695149986342
let y = Math.acos(1); // 1
let z = Math.atan(1); // 0.7853981633974483
```

min() и max()

Функции min() и max() возвращают соответственно минимальное и максимальное значение из набора чисел:

```
let max = Math.max(19, 45); // 45
let min = Math.min(33, 24); // 24
```

Эти функции необязательно должны принимать два числа, в них можно передавать и большее количество чисел:

```
let max = Math.max(1, 2, 3, -9, 46, -23); // 46
```

```
random()
```

Функция `random()` возвращает случайное число с плавающей точкой из диапазона от 0 до 1:

```
let x = Math.random(); // например, 0.34934820
```

Константы

Кроме методов объект `Math` также определяет набор встроенных констант, которые можно использовать в различных вычислениях:

`Math.PI` (число π): 3.141592653589793

`Math.SQRT2` (квадратный корень из двух): 1.4142135623730951

`Math.SQRT1_2` (половина от квадратного корня из двух): 0.7071067811865476

`Math.E` (число e или число Эйлера): 2.718281828459045

`Math.LN2` (натуральный логарифм числа 2): 0.6931471805599453

`Math.LN10` (натуральный логарифм числа 10): 2.302585092994046

`Math.LOG2E` (двоичный логарифм числа e): 1.4426950408889634

`Math.LOG10E` (десятичный логарифм числа e): 0.4342944819032518

Используем константы в вычислениях:

```
let x = Math.log(Math.E); // 1
```

```
let z = Math.tan(Math.PI/4); // 0.9999999999999999
```