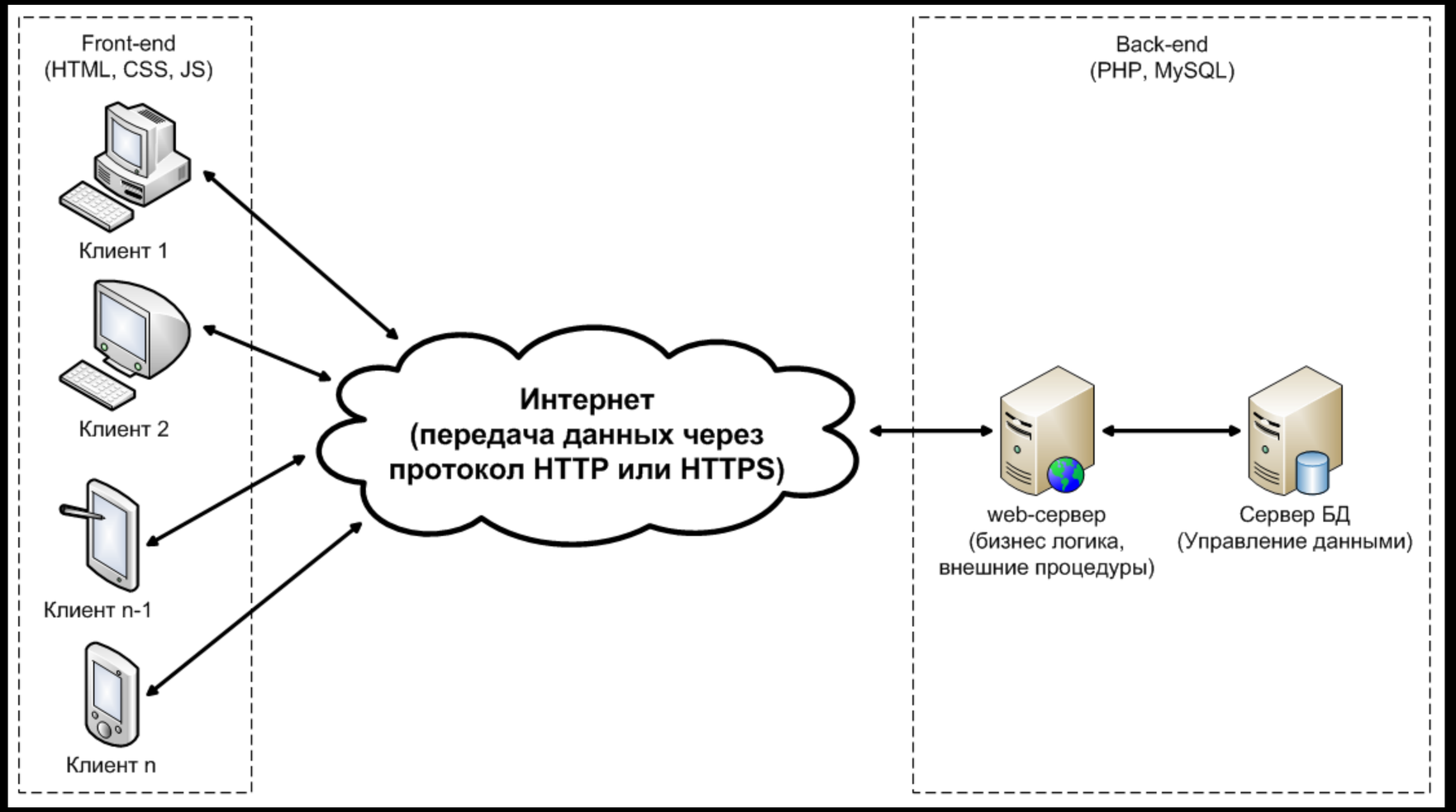
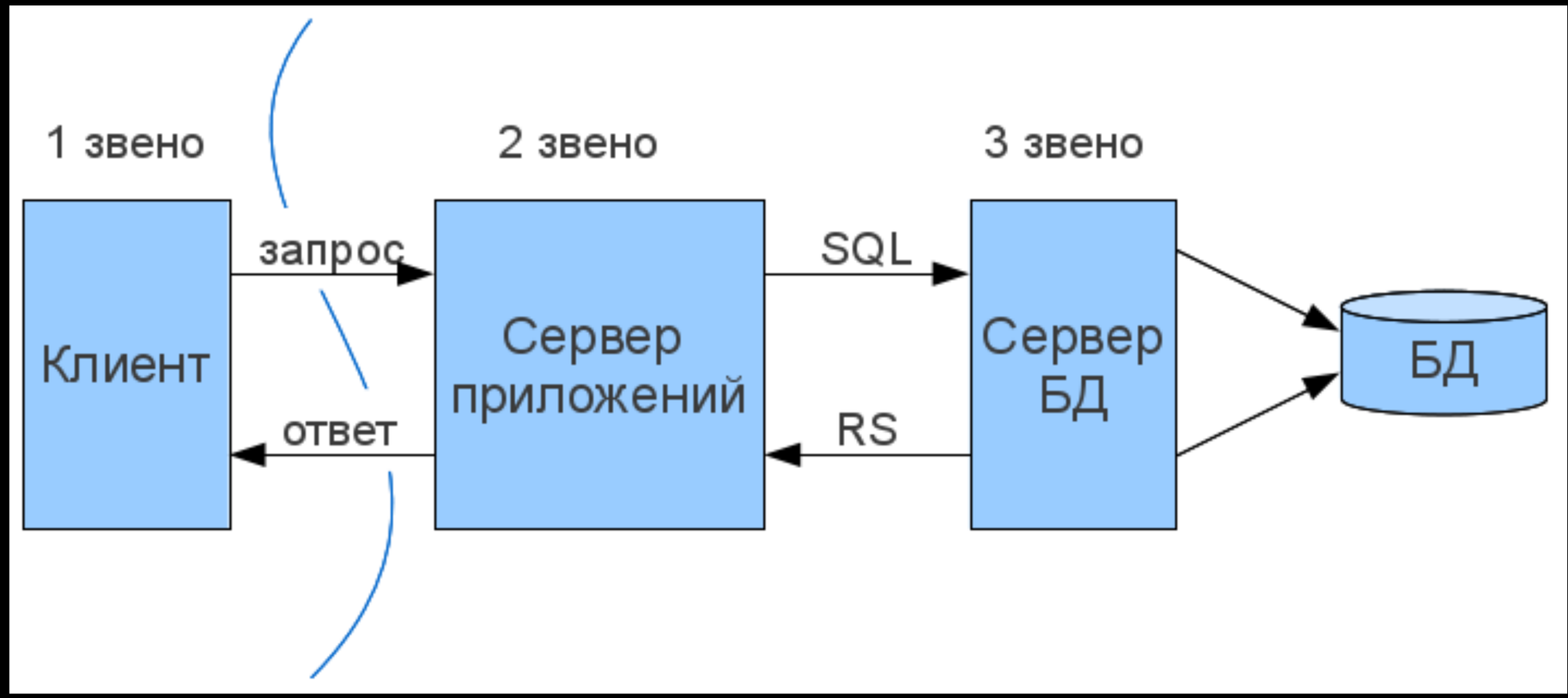
JavaScript

Сценарный язык программирования

Введение в web



Клиент-серверная архитектура



Внедрение скрипта в тело HTML

- Программы на языке JavaScript можно вставить в HTML при помощи тега SCRIPT
- <script>Сценарий</script>
  - type = text/javascript
  - language = JavaScript
- Отображается часть документа до script
- Исполняет его содержимое
- Отображается оставшаяся часть

Внешние скрипты

- <head>
  <script async src = “path/file.js”></script>
  </head>
- Атрибуты async и defer включают асинхронный режим работы
- Разница между async и defer:
  - defer сохраняет последовательность скриптов
  - defer ждёт, пока весь HTML-документ будет готов

Комментарии

- JS поддерживает однострочные и многострочные комметарии
  - // комментарий - однострочный комментарий
  - /* комментарий */ - многострочный комментарий

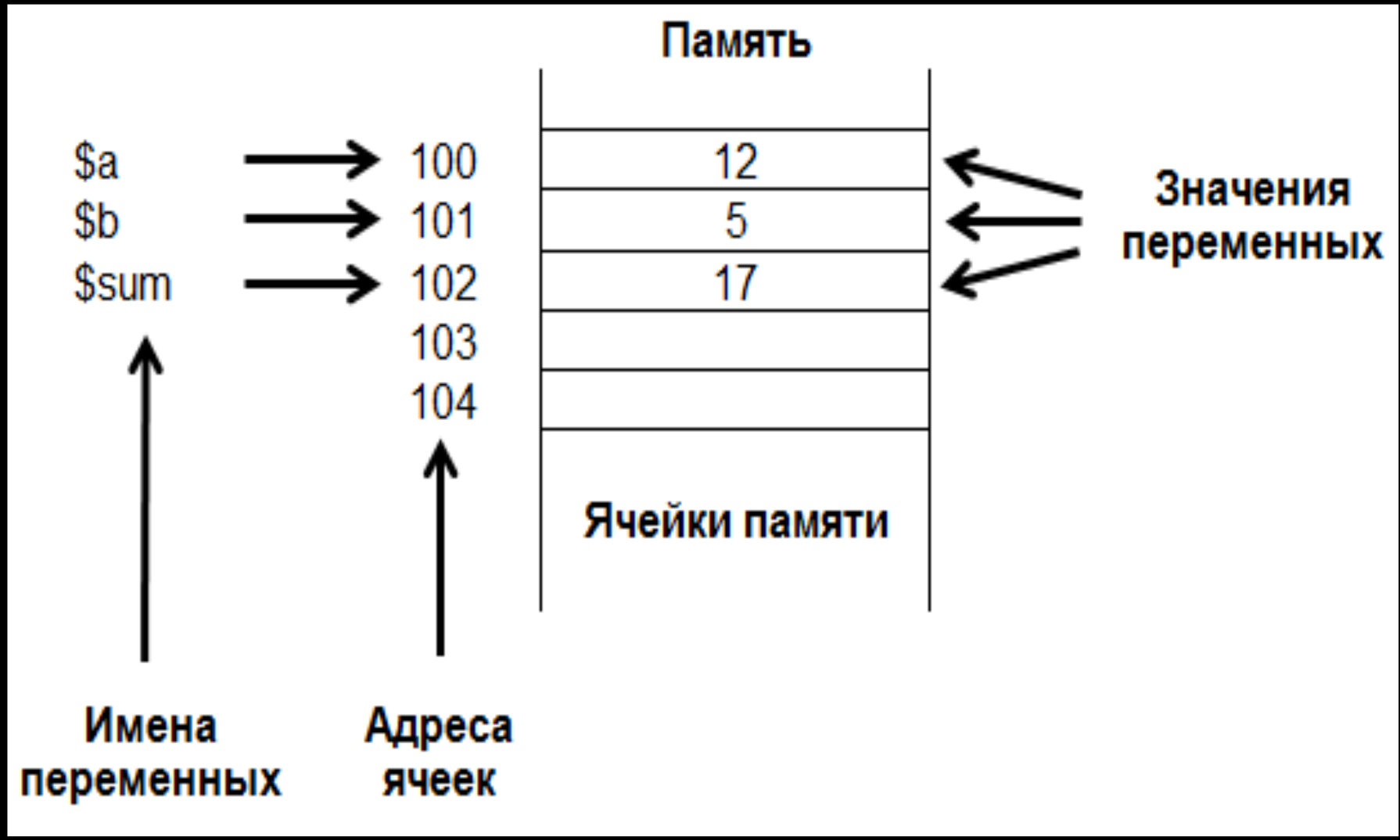
Стандарт ES6

- Стандарт EcmaScript 6 (ES6), добавил новые возможности и внёс в язык ряд исправлений
- Директива «use strict» переводит код в режим полного соответствия современному стандарту

Переменные

- *Переменная* – поименованная область памяти, адрес которой можно использовать для осуществления доступа к данным и изменять значение в ходе выполнения программы
- Для *объявления переменной* используется ключевое слово var
- Операция присвоения значений переменной осуществляется через оператор “=”

Переменные


Память
$a
$b
$sum
100
101
102
103
104
12
5
17
Значения переменных
Ячейки памяти
Имена переменных
Адреса ячеек

Имена переменных

- На имя переменной в JavaScript наложены всего два ограничения:
  - Имя может состоять из букв, цифр, ‘$’ и ‘_’
  - Первый символ не должен быть цифрой
- Корректно
  - var name;
  - var test15;
- Некорректно
  - var 1st;
  - var my.var;

Зарезервированные имена

break	default	return	var
case	delete	if	switch
void	return	catch	do
in	this	while	const
else	instanceof	throw	with
continue	finally	let	try
debugger	for	new	typeof

Типы данных

- В JavaScript существует 7 типов данных:
  - boolean
  - number
  - string
  - NULL
  - undefined
  - object
  - function

boolean

- Логический тип данных
- Принимает всего 2 значения
  - TRUE – истина – 1
  - FALSE – ложь – 0

- var checked = true;
- checked = false;

number

- Числовой тип данных
- Имеет несколько псевдотипов
  - NaN (Not a Number) – не число
  - Infinity – бесконечность

- var num = 10; // number
- num = 0.8; // number
- var inf = 1 / 0; // infinity
- var a = “text” * 5 // NaN

Способы записи

- Системы счисления
  - Двоичная 0b100   // 4
  - Шестнадцатиричная 0xFF // 255
  - Восьмиричная 010 // 8
- Экспоненциальная запись
  - var a = 3e5; // a = 300000;
  - var b = 3e-5; //b = 0.00003;

string

- Строковый тип данных
- В JavaScript одинарные и двойные кавычки равноправны

- var str1 = “Hello world!”;
- var str2 = ‘processing….’;
- var str3 = “str1 + str2”;

NULL

- Специальный тип данных состоящий из одного значения NULL
- Несет в себе смысловую нагрузка типа «ничего», «значение неизвестно»

- var age = null;

undefined

- Специальный тип данных состоящий из одного значения undefined
- Несет в себе смысловую нагрузка типа «значение не определено»

- var x; // undefined

object

- Объектный тип данных
- Язык имеет в своей структуре ошибку, по которой типы NULL и function определяются как тип данных object

- var user = { name: "Вася" }; // object

Оператор typeof

- Оператор typeof возвращает тип аргумента
- Поддерживает 2 синтаксические записи:
  - typeof x
  - typeof(x)

- typeof undefined // "undefined"
- typeof 1 // "number"
- typeof true // "boolean"
- typeof "foo" // "string"
- typeof {} // "object"
- typeof null // "object"
- typeof function(){} // "object "

Операторы

- *Операнд (*аргумент оператора*)* — то, к чему применяется оператор
- *Унарный оператор* — применяется к одному операнду
- *Бинарный оператор* — применяется к двум операндам

- `x = -a + b * c - 10;`

Арифметические

П	Н	Результат
-a	Отрицание	Смена знака a
a + b	Сложение	Сумма a и b
a – b	Вычитание	Разность a и b
a * b	Умножение	Произведение a на b
a / b	Деление	Частное от деления a на b
a % b	Остаток	Целочисленный остаток от
()	Скобки	Изменение приоритетов

Строковые

Пример	Название	Результат
a + b	Сложение	Строка склеенная из строк a и b
+a	Преобразование к числу	Строка a преобразуется в число

```
var a = “Hello”;
var b = ‘world’;
var c = a + “ ” + b + “!”;    // Hello world!

var d = “7”;
alert(+a);                    // число 7
```

Специальные символы

- \n – перевод строки
- \r – возврат каретки
- \f – перевод страницы
- \t – знак табуляции
- \’ – апостроф
- \” – кавычка
- \\ - обратный слэш

Логические

Пример	Название	Результат
!a	Отрицание	TRUE если a FALSE
a && b	Логическое	TRUE если a и b TRUE
a || b	Логическое	TRUE если a или b TRUE

Сравнения

Пример	Название	Результат
a == b	Равно	TRUE если a равно b
a === b	Тождественно	TRUE если a равно b и имеет тот же
a != b	Не равно	TRUE если a не равно b
a !== b	Тождественно не	TRUE если a не равно b или они имеют
a < b	Меньше	TRUE если a строго меньше b
a > b	Больше	TRUE если a строго больше b
a <= b	Меньше или	TRUE если a меньше или равно b
a >= b	Больше или	TRUE если a больше или равно b

Побитовые

Пример	Название	Результат
a & b	Побитовое 'и'	Устанавливаются только те биты, которые установлены и в a, и в b.
a | b	Побитовое или	Устанавливаются те биты, которые установлены либо в a, либо в b.
a ^ b	Исключающее или	Устанавливаются только те биты, которые установлены либо только в a, либо только в b
~ a	Отрицание	Устанавливаются те биты, которые в a не установлены, и наоборот.
a << b	Сдвиг влево	Все биты переменной a сдвигаются на b позиции влево (знак сохраняется)
a >> b	Сдвиг вправо	Все биты переменной a сдвигаются на b позиции вправо (знак сохраняется)
a >>> b	Сдвиг вправо с заполнением	Все биты a сдвигаются на b позиций вправо, добавляя нули слева

Инкремент и декремент

- Инкремент – увеличивает значение переменной на единицу
- Декремент – уменьшает значение переменной на единицу
- ++a – Префиксный инкремент
- a++ – Постфиксный инкремент
- --a – Префиксный декремент
- a-- – Постфиксный декремент
- a = 10;
- alert(a++);
- alert(++a);

Взаимодействие с пользователем

- **alert** выводит сообщение
  alert(“Hello!”);
- **prompt** выводит сообщение и ждёт, пока пользователь введёт текст
  var answer = prompt(“2 + 2 = ”,“”);
- **confirm** выводит сообщение и ждёт, пока пользователь нажмёт «OK» или «CANCEL»
  var admin = confirm(“Admin?”);

Работа с консолью

- Функция console выводит сообщение в консоль
  - warn – предупреждение
  - info – информация
  - dir - развертка

Математические функции

- Math.acos
- Math.asin
- Math.atan
- Math.atan2
- Math.exp
- Math.min
- Math.random
- Math.sqrt
- Math.log
- Math.round
- Math.floor
- Math.ceil
- Math.sin
- Math.cos
- Math.tan
- Math.pow
- Math.max
- Math.abs

Математические константы

- Math.E
- Math.LN2
- Math.LOG2E
- Math.LOG10E
- Math.PI
- Math.SQRT12
- Math.SQRT2
- Math.LN10

Логическое преобразование

- Функция Boolean() служит для явного логического преобразования значения
- False
  - undefined, null
  - 0, NaN
  - ''
- True
  - Все остальные

Численное преобразование

- Функция Number() служит для явного численного преобразования значения

undefined	NaN
null	0
true / false	1 / 0
Строка	Пробелы по краям обрезаются '' – 0 Число – число Иначе - NaN

Строковое преобразование

- Функция String() служит для явного строкового преобразования значения
- Значение преобразуется в строку «как есть»

Мягкое преобразование

- Функции ParseInt() и ParseFloat() позволяют преобразовать значение к числу «последовательно»
- Функция ParseInt() переводит число из указанной системы счисления

Проверка типов

- isNaN() -проверяет на NaN
- isFinite() – проверяет, является ли число числом
- isNumeric() – проверяет на тип данных number

Неточные вычисления

- Из-за стандарта IEEE 754 на число выделяется ровно 8 байт
- В связи с этим при работе с некоторыми значениями могут возникать погрешности
  - console.log(0.1+0.2); //0.30000000000000004
- Метод n.toFixed() округляет число до указанного знака и приводи к строковому значению (дополняет нулями по необходимости)