**3 РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ**

**3.1 Zenity**

Утилита **zenity** – это средство создания диалоговых окон в режиме командной строки. Следует отметить, что на самом деле диалоговые окна создаются средствами Gtk+, поэтому в системе должны быть установлены соответствующие библиотеки.

Чтобы начать использовать **zenity** на практике не требуется обладать особыми знаниями или умениями, достаточно просто познакомиться с различными опциями (ключами), позволяющими в полной мере использовать возможности этой программы.

В zenity определены четыре типа диалоговых окон для вывода сообщений:

ошибка (ключ --error);

информация (ключ --info);

вопрос (ключ --question);

предупреждение (ключ --warning).

Установка zenity: sudo apt-get install zenity

Теперь, если выполнить команду zenity, то выведется следующая ошибка:

You must specify a dialog type. See 'zenity --help' for details

Поэтому для того, чтобы изучить полный список параметров (и, соответственно, возможностей zenity), можно использовать именно эту команду.

Для того, чтобы создать простейшее диалоговое окно, необходимо выполнить команду:

zenity --entry --title='addition' --text='enter first number'

Для того, чтобы эту же команду выполнить из файла с таким же результатом, необходимо:

создать файл с расширением .sh, например, add.sh

добавить в него следующие строки:

#!/bin/sh

num1=$(zenity --entry --title='addition' --text='enter first number')

echo $num1

echo '+'

num2=$(zenity --entry --title='addition' --text="enter second number")

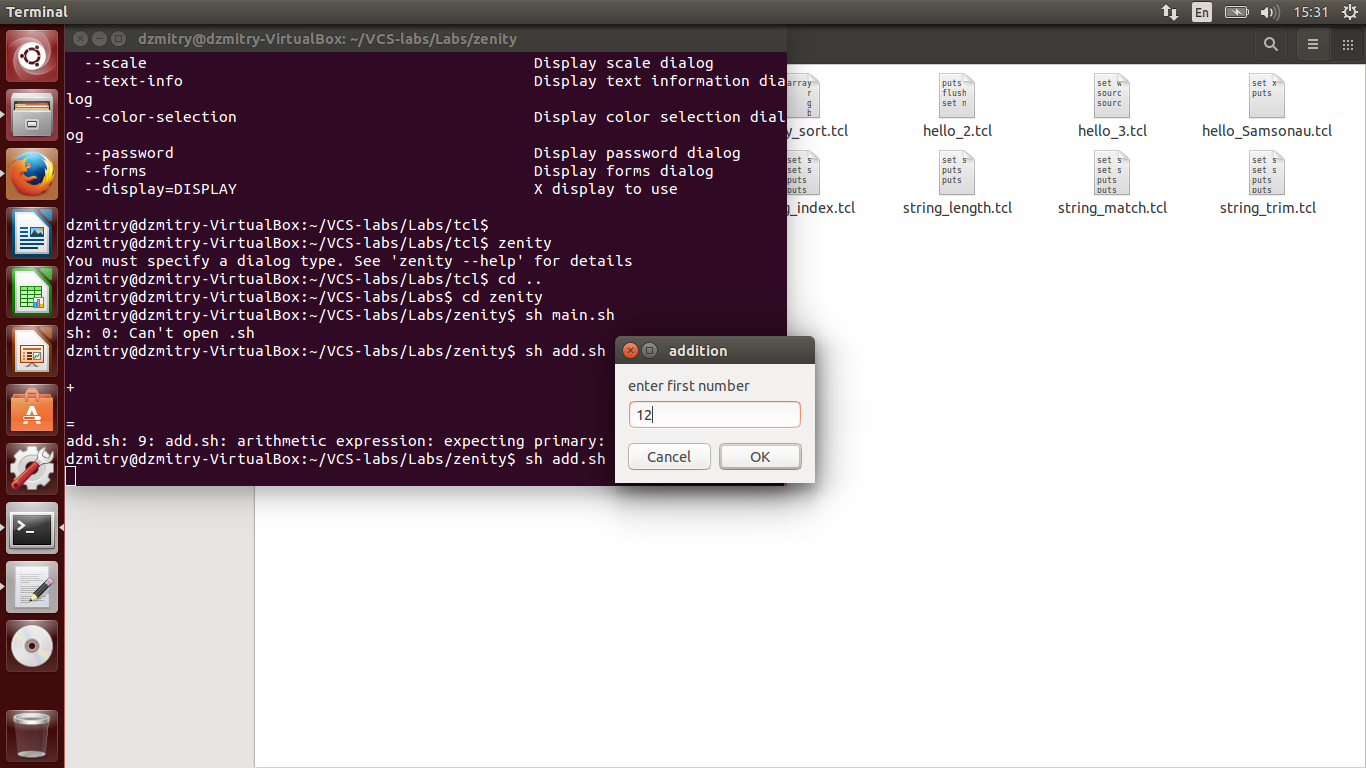
echo $num2

echo '='

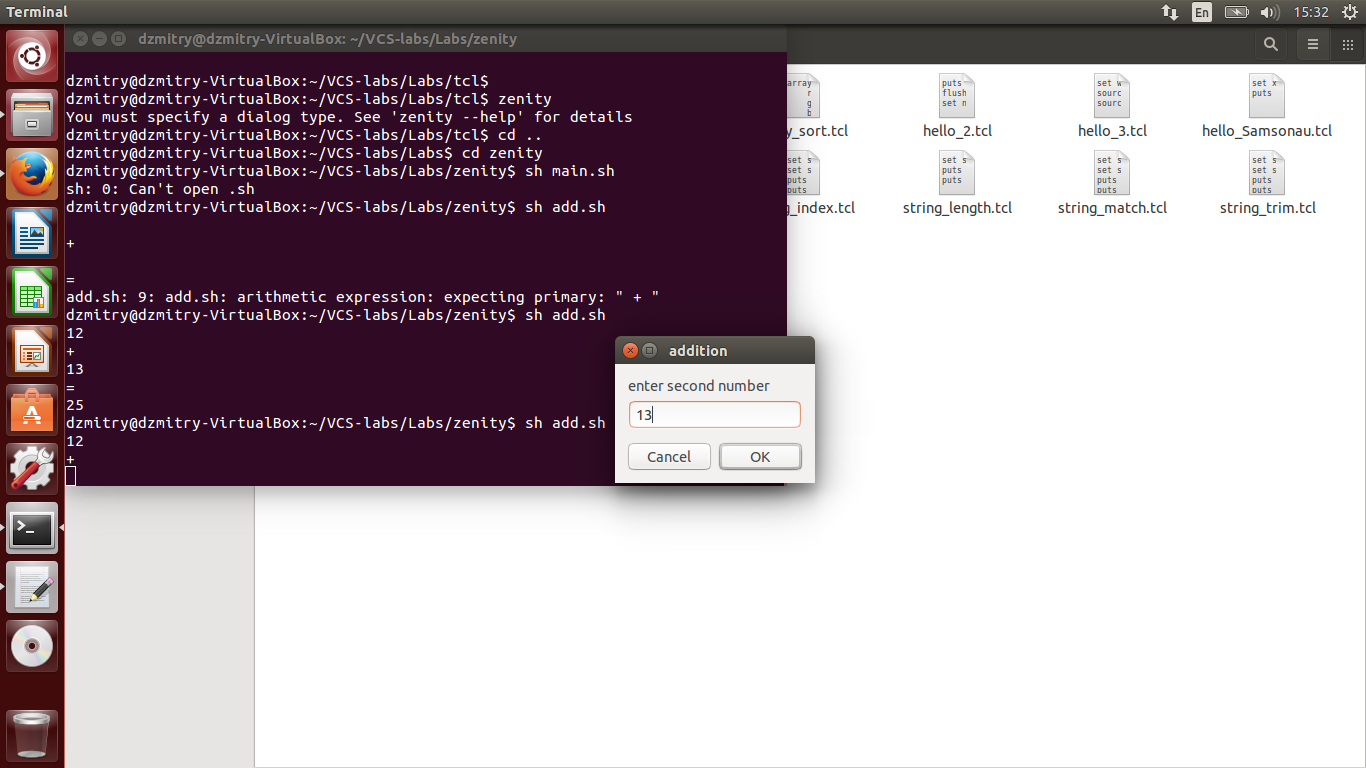
echo $(($num1 + $num2))

выполнить в консоли sh add.sh

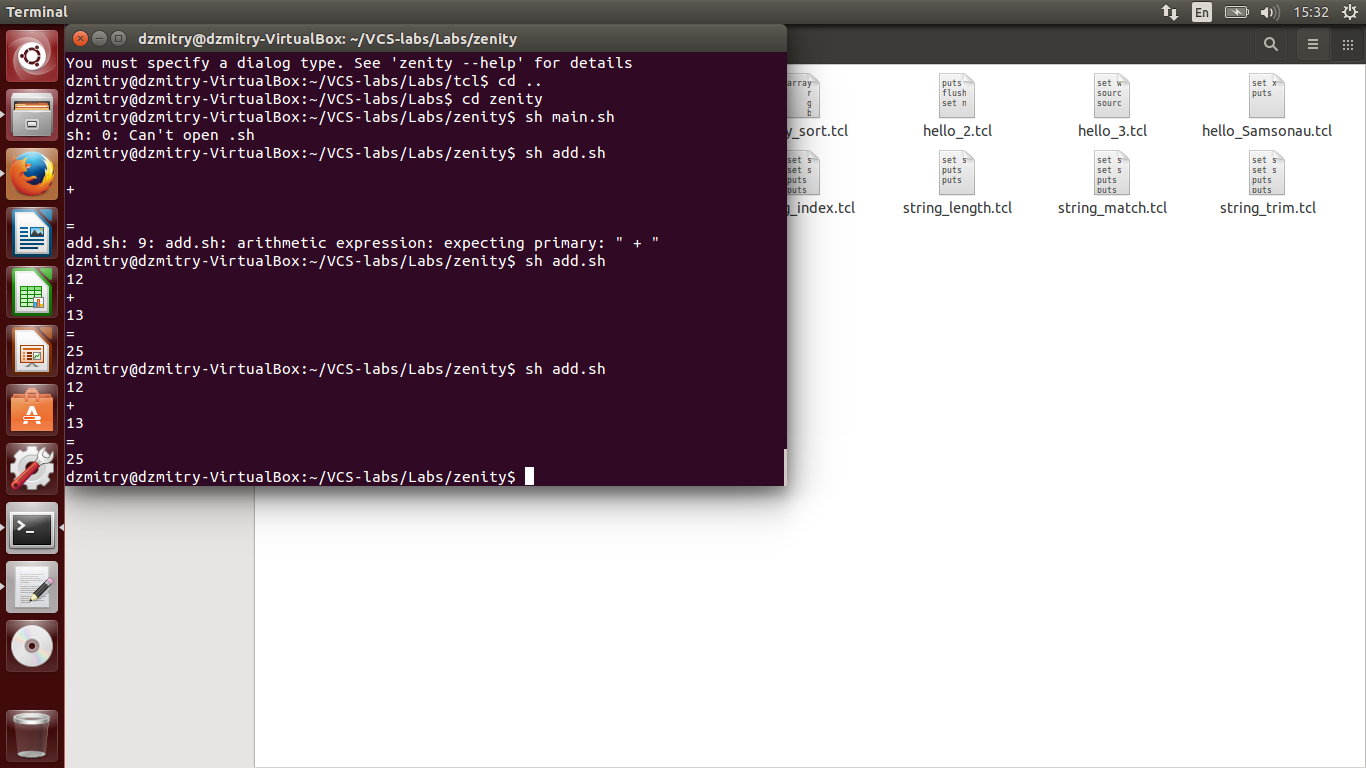
Если вы всё сделали правильно, то отобразится следующее окно:



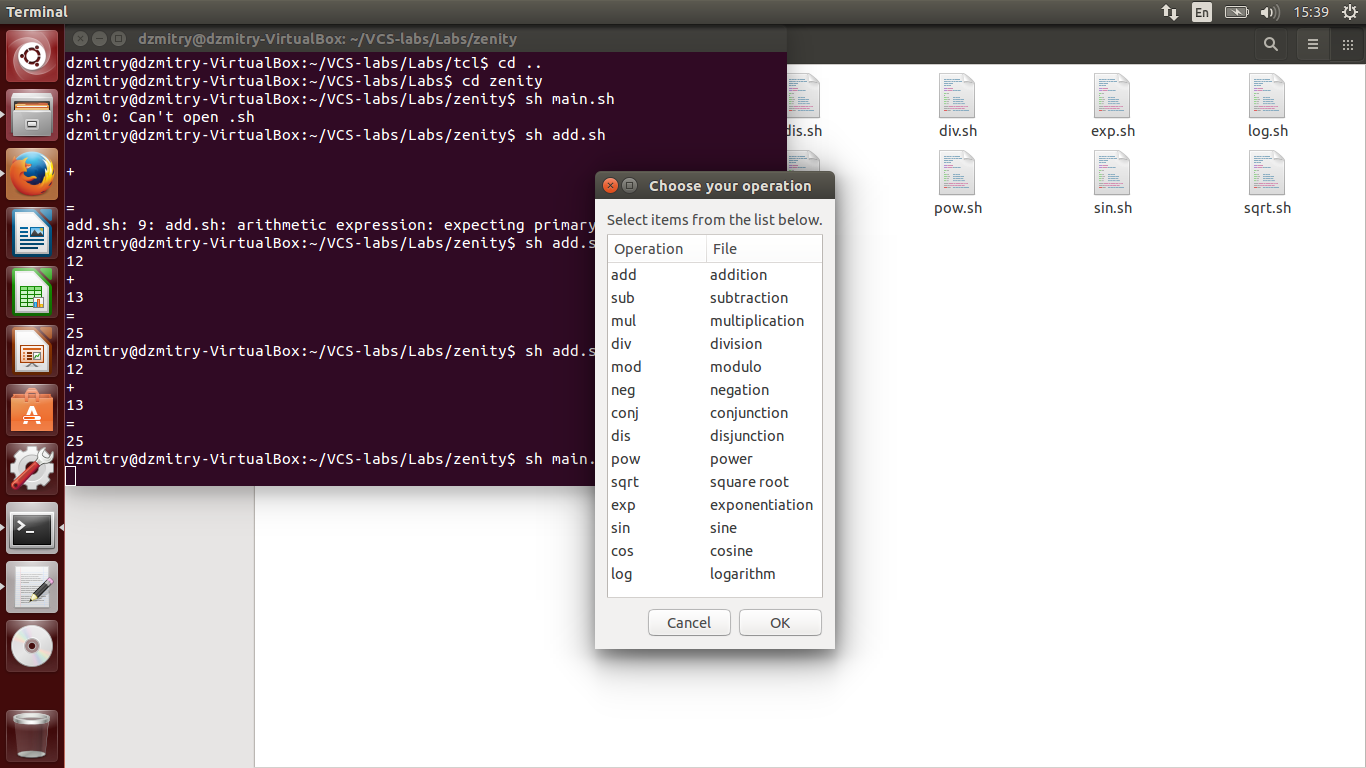
После этого следующее



И в консоли вы увидите результат выполнения программы.



\*:Для создания файла main.sh воспользуйтесь следующей структурой файла:

#!/bin/sh

operation=$(

zenity --list \

…

…

add addition \

sub subtraction \

...

…

…

)'.sh'

sh $operation

Первый пропуск необходимо будет заполнить строками с указанием названия и ширины-высоты окна. Попробуйте получить помощь по всем функциям zenity при помощи --help. Второй пропуск заполните в соответствии с названиями всех компонентов, которые будут присутствовать в вашем решении.

\*: Для создания особых функций попробуйте следующие команды:

echo "c($num)" | bc -l

echo "s($num)" | bc -l

echo "l($num)" | bc -l

\*: Для создания функции возведения в степень попробуйте цикл:

pow=$num1

for i in $(seq 2 …)

do

pow=$(($pow\*…))

done

вам необходимо заполнить два пропуска некими значениями.

\*\*: Функция извлечения корня является нетривиальной; изучите следующий код (он правильный) и заставьте его работать, правильно инициализировав значения:

while [ $(($prev-$b)) -ge 1 ]; do

prev=$b;

b=$((($b+$num/$b)/2));

done

**3.2 tcl**

Для выполнения задания вам понадобится tcl – скриптовый язык высокого уровня. Для того, чтобы запустить программу на tcl, её не нужно компилировать – достаточно вызвать её при помощи утилиты командной строки tlcsh (соответвенно tcl и sh -- шелл):

dzmitry@dzmitry-VirtualBox:~/VCS-labs/Labs/tcl$ tclsh array\_sort.tcl

Основные команды языка:

|  |  |
| --- | --- |
| Пример функции | Краткое описание |
| set x "Hello, World!" | Присваивание строки переменной х |
| puts "A simple string: $x\n" | Вывод на экран |
| source [file join $where hello.tcl] | Добавить .tcl файл в другой .tcl файл |
| set together [concat $first $second] | Объединить массивы |
| array set copy [array get colors] | Скопировать значение |
| unset colors(red) | Удалить из массива |
| string compare s1 s2 | Сравнить строки |
| if {… } {…} | Если-тогда выражение |
| append result "World" | Объединение строк |
| format "%f" 43.5 | Форматирование |
| string index $s1 0 | Первое вхождение в строку |
| string last $s2 $s1 | Последнее вхождение в строку |
| string length $s1 | Длина строки |
| string match "\*@\*.com" $s1 | Проверка на паттерн |
| string match {tcl} $s1 | Проверка на присутствие строки |
| string wordend $s1 20 | Получить конец слова, начиная с текущей позиции |
| string wordstart $s1 20 | Получить начало слова, начиная с текущей позиции |
| string trimright $s1 $s2 | Убрать ненужные символы справа |
| string trimleft $s1 $s2 | Убрать ненужные символы слева |
| string trim $s1 $s2 | Убрать ненужные символы с двух сторон |
| foreach name [array names copy] {  puts "$name is $copy($name)"  } | Вывод массива на экран |

**3.3 html**

HTML (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) HyperText Markup Language — «язык [гипертекстовой](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) разметки») — стандартный [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) документов во [Всемирной паутине](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0). Большинство [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) содержат описание разметки на языке HTML (или[XHTML](https://ru.wikipedia.org/wiki/XHTML)). Язык HTML интерпретируется [браузерами](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML является приложением [SGML](https://ru.wikipedia.org/wiki/SGML) (стандартного обобщённого языка разметки) и соответствует международному стандарту [ISO](https://ru.wikipedia.org/wiki/ISO) 8879.

Язык [XHTML](https://ru.wikipedia.org/wiki/XHTML) является более строгим вариантом HTML, он следует всем ограничениям [XML](https://ru.wikipedia.org/wiki/XML) и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам [HTTP](https://ru.wikipedia.org/wiki/HTTP) или[HTTPS](https://ru.wikipedia.org/wiki/HTTPS), в виде простого текста или с использованием [шифрования](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <TABLE> предназначен для создания в документах таблиц, но часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

HTML — теговый язык разметки [документов](https://ru.wikipedia.org/wiki/HTML-%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82). Любой документ на языке HTML представляет собой набор [элементов](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B_HTML), причём начало и конец каждого элемента обозначается специальными пометками — [тегами](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%B3_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8)). Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки <br>). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, размер шрифта для элемента font). Атрибуты указываются в открывающем теге. Вот примеры фрагментов HTML-документа:

<strong>Текст между двумя тегами — открывающим и закрывающим.</strong>

<a href="http://www.example.com">Здесь элемент содержит атрибут href, то есть гиперссылку.</a>

А вот пример пустого элемента: <br>

Регистр, в котором набрано имя элемента и имена атрибутов, в HTML значения не имеет (в отличие от XHTML). Элементы могут быть вложенными. Например, следующий код:

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

</head>

<body>

<p>

<b>

Этот текст будет полужирным,

<i>а этот - ещё и курсивным</i>

</b>

</p>

</body>

</html>

даст такой результат:

Этот текст будет полужирным, а этот — ещё и курсивным

Кроме элементов, в HTML-документах есть и сущности ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) entities) — [«специальные символы»](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%B5%D0%BC%D0%BE%D0%BD%D0%B8%D0%BA%D0%B8_%D0%B2_HTML). Сущности начинаются с символа [амперсанда](https://ru.wikipedia.org/wiki/%D0%90%D0%BC%D0%BF%D0%B5%D1%80%D1%81%D0%B0%D0%BD%D0%B4) и имеют вид&имя; или &#NNNN;, где NNNN — код символа в [Юникоде](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) в десятичной системе счисления.

Например, &copy; — знак [авторского права](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%BE) (©). Как правило, сущности используются для представления символов, отсутствующих в кодировке документа, или же для представления «специальных» символов: &amp; — амперсанда (&), &lt; — символа «меньше» (<) и &gt; — символа «больше» (>), которые некорректно записывать «обычным» образом, из-за их особого значения в HTML.

Подробнее по этой теме см.: [Элементы HTML](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B_HTML).

Подробнее по этой теме см.: [Википедия:Специальные символы](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D1%8B).

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML <!DOCTYPE…>, которая обычно выглядит примерно так:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">

Если эта строка не указана, то добиться корректного отображения документа в браузере становится труднее.

Далее обозначается начало и конец документа тегами <html> и </html> соответственно. Внутри этих тегов должны находиться теги заголовка (<head></head>) и тела (<body></body>) документа.

Варианты DOCTYPE для HTML 4.01

Строгий (Strict): не содержит элементов, помеченных как «устаревшие» или «не одобряемые» (deprecated).

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

Переходный (Transitional): содержит устаревшие теги в целях совместимости и упрощения перехода со старых версий HTML.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

С [фреймами](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC_(HTML)) (Frameset): аналогичен переходному, но содержит также теги для создания наборов фреймов.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"

"http://www.w3.org/TR/html4/frameset.dtd">

Варианты DOCTYPE для HTML 5

В [HTML 5](https://ru.wikipedia.org/wiki/HTML_5) используется только один вариант DOCTYPE:

<!DOCTYPE html>