

# NodeJS



JavaScript  
Courses

[vk.com/js.courses](https://vk.com/js.courses)

[js.courses.dp.ua/files](https://js.courses.dp.ua/files)

# NodeJS



<https://nodejs.org/en/>

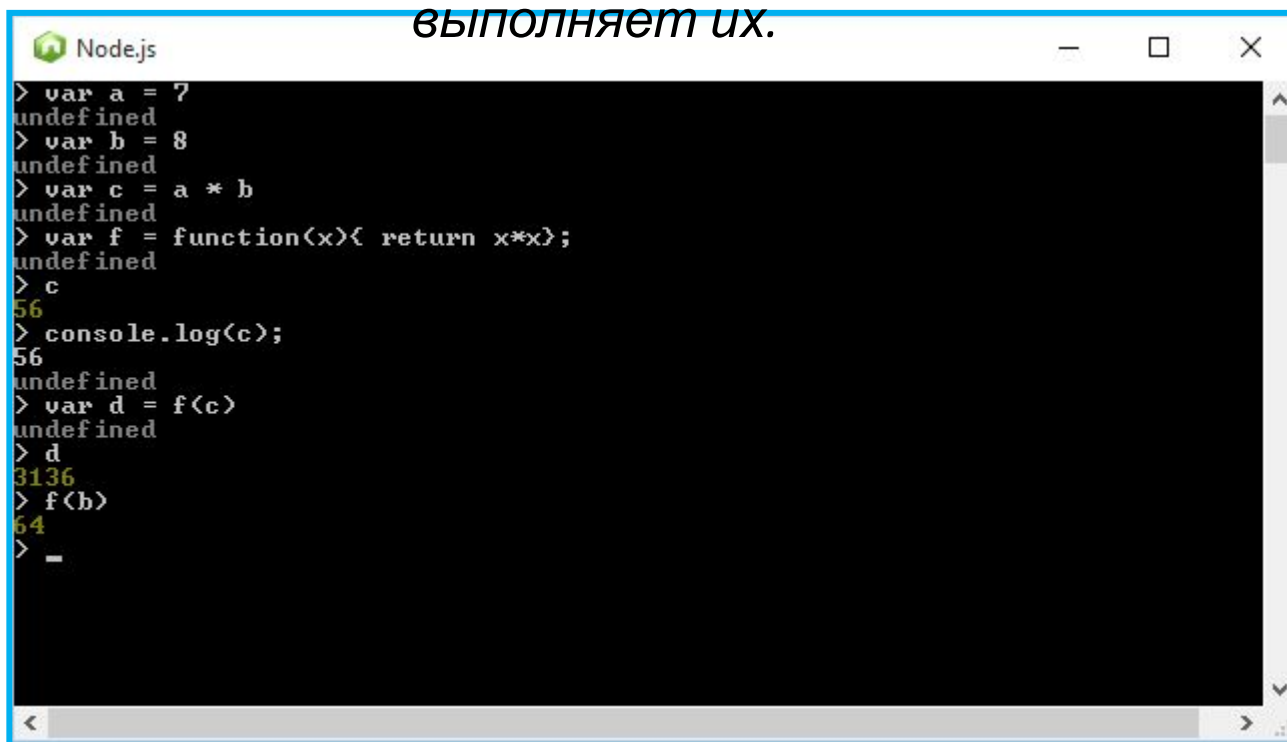
**NodeJS** – программа, которая ставится на компьютер и позволяет выполнять код на языке **JavaScript** прямо на вашем компьютере (с полным доступом к компьютеру, без ограничений браузера, да и без самого браузера). Т.е. **NodeJS** превращает **JavaScript** в язык общего назначения, ставит его в один ряд с C#, Java, Ruby, Python и прочими.

**Скачиваем и  
устанавливаем**

# Интерпретатор JavaScript-

## кода

*В составе Node.js входит программа-интерпретатор, которая разбирает текстовые файлы с кодом и выполняет их.*

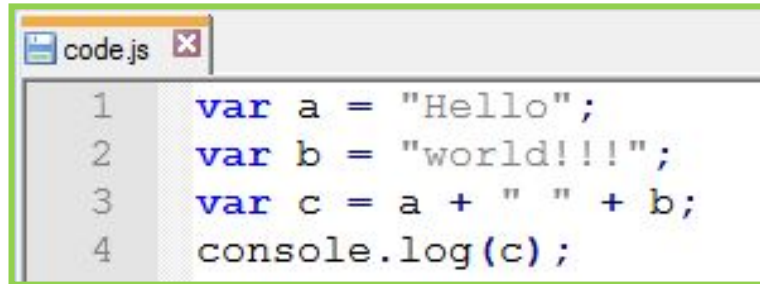
A screenshot of the Node.js REPL (Read-Eval-Print Loop) window. The window has a title bar with the Node.js logo and the text "Node.js". The background is black with white text. The code being executed is as follows:

```
> var a = 7
undefined
> var b = 8
undefined
> var c = a * b
undefined
> var f = function(x){ return x*x; }
undefined
> c
56
> console.log(c);
56
undefined
> var d = f(c)
undefined
> d
3136
> f(b)
64
> -
```

*Позволяет проверить JavaScript код, и сразу получить результат его работы построчно.*

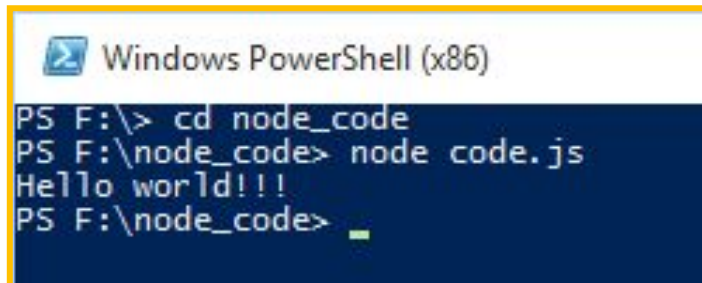
# Интерпретатор JavaScript-

Создайте файл **кода**  
*code.js*



```
1 var a = "Hello";  
2 var b = "world!!!";  
3 var c = a + " " + b;  
4 console.log(c);
```

Выполните в консоли команду **node** *code.js* (вместо *code.js* может быть путь к любому js-файлу).



```
Windows PowerShell (x86)  
PS F:\> cd node_code  
PS F:\node_code> node code.js  
Hello world!!!  
PS F:\node_code> _
```

Консольная команда **node** *code.js*  
позволяет запустить на выполнение  
файл с JS-кодом целиком.

# NodeJS

***Node** работает на стороне сервера (непосредственно на компьютере, без песочниц типа браузера), для **Node** не существует **DOM** и прочей инфраструктуры браузера, т.е. никаких функций **prompt()**, **alert()** не существует (к сожалению).*

*Нет и объекта **XMLHttpRequest**, но это не мешает **Node** загружать данные из сети. Но в **Node** есть модули...*

# Модули в NodeJS

```
load.js
1  var request_function = require('request');
2
3  request_function('http://www.lib.ru', function (error, response, body) {
4      console.log(body)
5  });
```

Запускаем  
*node load.js*

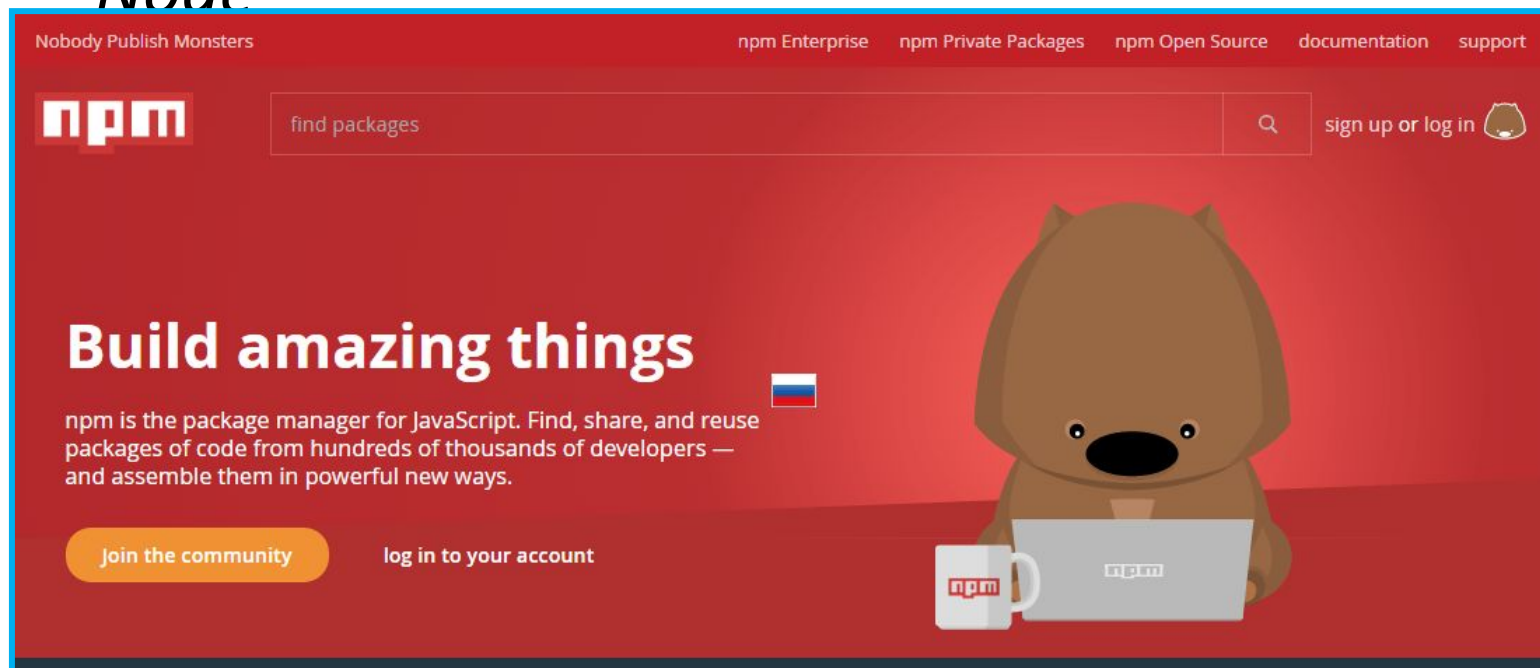
```
Windows PowerShell (x86)
PS D:\> cd F:\node
PS F:\node> node load.js
module.js:329
  throw err;
  ^
Error: Cannot find module 'request'
    at Function.Module._resolveFilename (module.js:327:15)
    at Function.Module._load (module.js:278:25)
    at Module.require (module.js:355:17)
    at require (internal/module.js:13:17)
    at Object.<anonymous> (F:\node\load.js:1:24)
    at Module._compile (module.js:399:26)
    at Object.Module._extensions..js (module.js:406:10)
    at Module.load (module.js:345:32)
    at Function.Module._load (module.js:302:12)
    at Function.Module.runMain (module.js:431:10)
PS F:\node> _
```

*Однако не все модули идут в комплекте с Node....*

*Однако есть NPM*

# NPM –Node Package Manager

*Всемирная библиотека модулей для  
Node*



<https://www.npmjs.com/>

*Однако не все модули идут в комплекте с  
Node....*

*Однако есть NPM....*

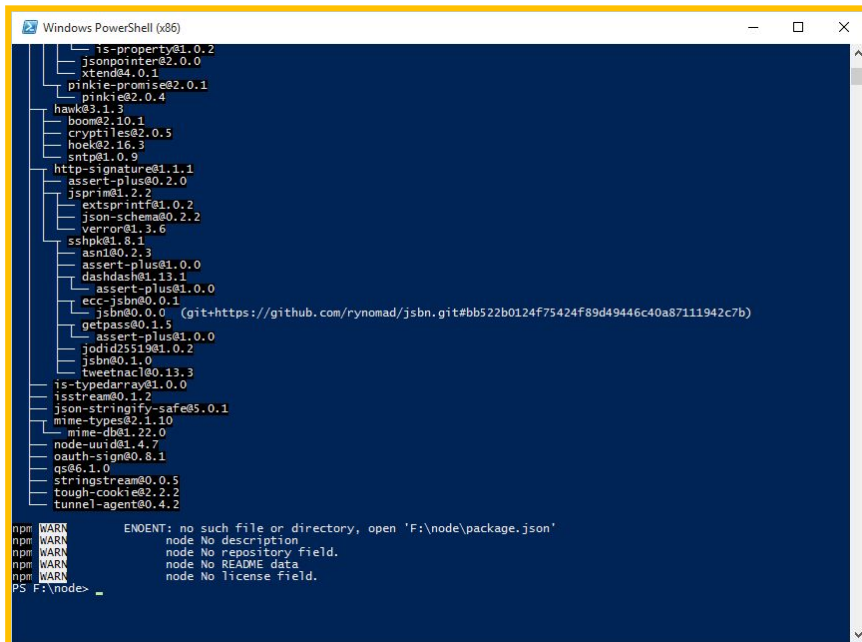
# NPM – Node Package Manager

*Когда модуля не хватает, то пишем:*

**npm install**

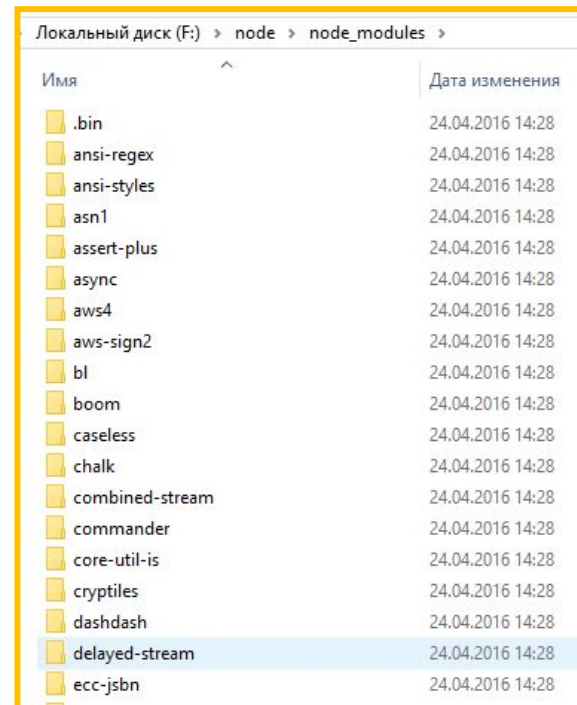
PS F:\> npm install **имя\_модуля** request

*И система управления пакетами установит в текущую папку требуемый модуль и все зависимые модули тоже.*



```
Windows PowerShell (x86)
1s-property@1.0.2
jsonpointer@2.0.0
xtend@4.0.1
pinkie-promise@2.0.1
pinkie@2.0.4
hawk@3.1.3
boom@2.10.1
cryptiles@2.0.5
hoek@2.16.3
sntp@1.0.9
http-signature@1.1.1
assert-plus@0.2.0
jsprim@1.2.2
extsprintf@1.0.2
json-schema@0.2.2
verror@1.3.6
sshpk@1.8.1
asn1@0.2.3
assert-plus@1.0.0
dashdash@1.13.1
assert-plus@1.0.0
ecc-jsbn@0.1.1
jsbn@0.0.0 (git+https://github.com/rynomad/jsbn.git#bb522b0124f75424f89d49446c40a87111942c7b)
getpass@0.1.5
assert-plus@1.0.0
jodid25519@1.0.2
jsbn@0.1.0
tweetnacl@0.13.3
1s-typedarray@1.0.0
1sstream@0.1.2
json-stringify-safe@5.0.1
mime-types@2.1.10
mime-db@1.22.0
node-uuid@1.4.7
oauth-sign@0.8.1
qs@6.1.0
stringstream@0.0.5
tough-cookie@2.2.2
tunnel-agent@0.4.2

npm WARN ENOENT: no such file or directory, open 'F:\node\package.json'
npm WARN node No description
npm WARN node No repository field.
npm WARN node No README data
npm WARN node No license field.
PS F:\node>
```



Имя	Дата изменения
.bin	24.04.2016 14:28
ansi-regex	24.04.2016 14:28
ansi-styles	24.04.2016 14:28
asn1	24.04.2016 14:28
assert-plus	24.04.2016 14:28
async	24.04.2016 14:28
aws4	24.04.2016 14:28
aws-sign2	24.04.2016 14:28
bl	24.04.2016 14:28
boom	24.04.2016 14:28
caseless	24.04.2016 14:28
chalk	24.04.2016 14:28
combined-stream	24.04.2016 14:28
commander	24.04.2016 14:28
core-util-is	24.04.2016 14:28
cryptiles	24.04.2016 14:28
dashdash	24.04.2016 14:28
delayed-stream	24.04.2016 14:28
ecc-jsbn	24.04.2016 14:28



# Модули в

```
1 var request_function = require('request');
2
3 request_function('http://www.lib.ru', function (error, response, body){
4     console.log(body)
5 });
```

Запускаем  
снова  
**node load.js**

```

PS F:\node> node load.js
<html><head><title>Lib.Ru: 0000000000 00000000 00000000</title></head><body><pre>
<center><h1><small><a href=http://lib.ru/>Lib.Ru</a></small> 0000000000 00000000 00000000<font size=-2>000 0000000000
00000000000000000000 00000000 00000000 00000000000000000000.</font></h1><font size=-1><form action=GrepSearch method=GET><b>
00000000</b></INPUT TYPE=TEXT NAME=Search size=9><input type=submit value=Search></font> <b><a href=/PROZA/>000000</a> <a
href=/INPROZ/>00000000</a> <a href=/POEZIQ/>00000000</a> <a href=/RUFANT/>0000000000</a> <a href=/RUSS DETEKTIV/>0000000000
</a> <a href=http://az.lib.ru/>00000000</a> <a href=/HISTORY/>00000000</a> &nbsp;<a href=/dir_StripDir.html>0 00.<small>
>>></small></a></b></form></center><center><a href=What-s-new><b>00000000</b></a></a><a href=/HITPARAD/><b>0000000000</b></a></a>
</a></a><a href=http://samlib.ru/>0000000000</a></a><a href=http://music.lib.ru/>00000000</a></a> <a href=http://www.artlib.ru/>000000
000</a></a><a href=http://world.lib.ru/>0000000000</a></a><a href=http://turizm.lib.ru/>00000000</a></a><a href=http://artofwar.rux
><b>ArtOfWar</b></a></a><a href=http://okopka.ru/>00000000</a></a><a href=/Forum/>00000000</a></a><br>000000000000 00000000 <a href=http:
//lit.lib.ru/>000000000000</a> <a href=http://fan.lib.ru/>000000000000</a> <a href=http://det.lib.ru/>0000000000000000</a> <a href
f=http://4put.ru/><font color=brown><b>000000000000</b></font></a></noindex> <a href=Mirrors><b>00000000</b></a></small> [
<b><a href=/koi/>koi</a></b></b><b><a href=/win/>win</a></b></b><b><a href=/lat/>lat</a></b></b></small> <img src=http://counter.rambler.ru/top100.cnt?59291 width=1 height=1 border=0><!--hr n
oshade size=1--></center></pre><font size=-1><b>23 aug 06. 5.5Gb</b></b><b>META NAME="keywords" CONTENT="CompuLib,LibMain,LibM
oshkow,LibAuthor">. 00000 0000000000 0 0000000 www-000000000000, 00000000 0 1994. 0000000 0 000000000
0000000000 0000000000 0. 0000000000000000000000000000000000, 0000000000 0 00000000, 00000000000000000000 0 0000,
00000000 0 0000000, 000 0 00000000 000, 0000000 0 0000000000, 0000000000 0 0000000000, 0 0.0. 0 0.0.<br></font>
<font size=-1>
<a href=AWARDS/>00000000</a></b><b>000-2003,POTOP, IT-100,000-2001</b></b>.
<br></font>
<font size=-2><b>00000000</b></b> <noindex><a href=http://lib.ru/>lib.ru</a>, <a href=http://www.kulichki.com/moshkow/>00000000</a> 00 <a href=http://www.kulichki.com/>kulichki.com</a>,

```

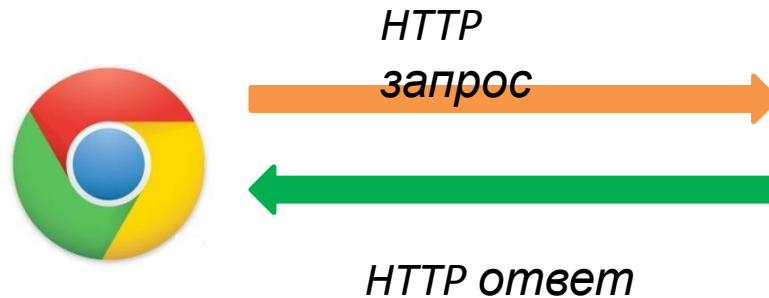
Теперь нужный модуль установлен в тот же каталог, в котором у нас лежит файл с кодом, и мы можем запускать скрипт

# Асинхронная и не блокирующая модель NodeJS, привыкаем....

```
code.js x
1  var stdin = process.openStdin();
2
3  console.log("Enter you name: ");
4
5  stdin.addListener("data", function(txt){
6      console.log("You enter: " + txt);
7  });
8
9  console.log("This text after start waiting.");
```

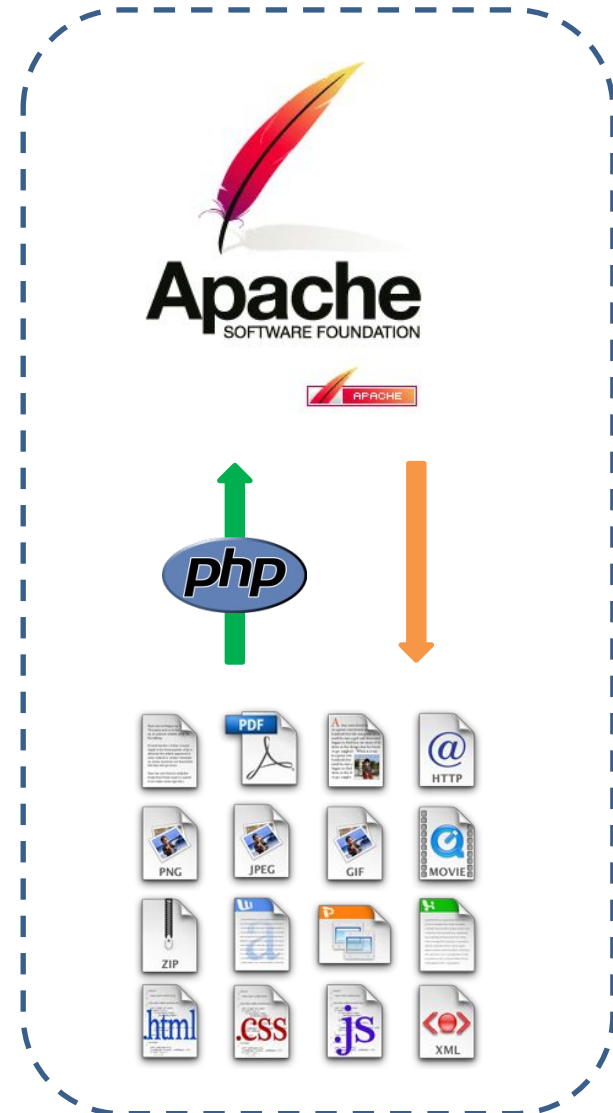
```
Windows PowerShell (x86)
PS F:\node_code> node code.js
Enter you name:
This text after start waiting.
Ivan
Enter name: Ivan
```

# HTTP-сервер (Web-сервер)

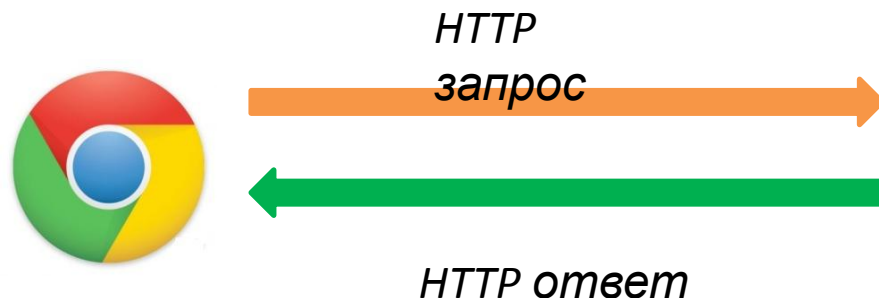


Основная задача **веб-сервера** ждать запросы от браузера и отправлять браузеру **веб-страницу** (и сопутствующие файлы) когда браузер их запросит.

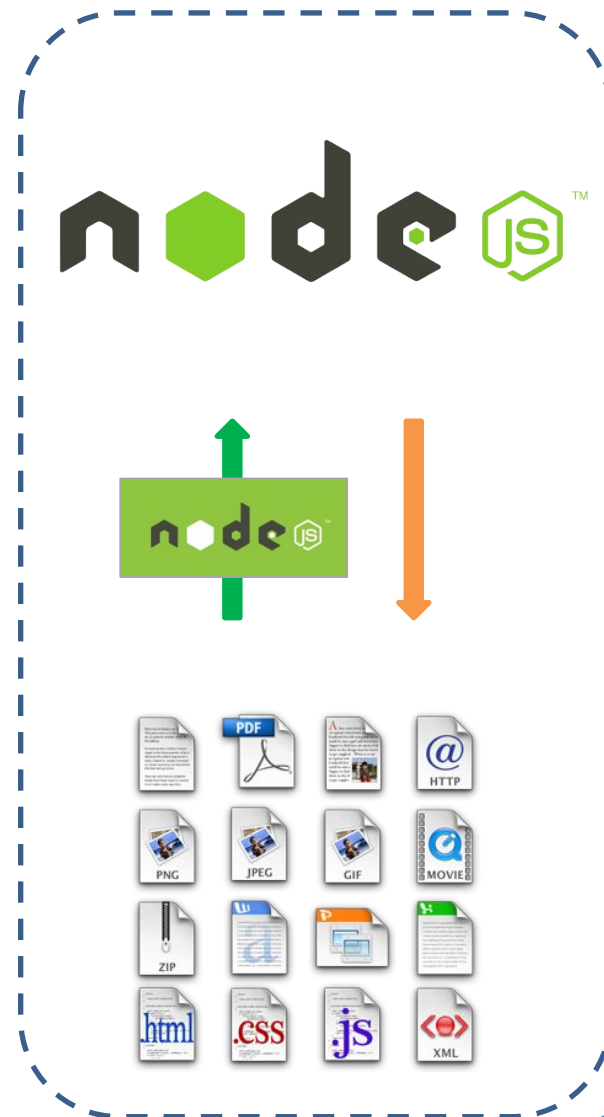
Перед отправкой файлов, сервер может их модифицировать, при помощи сценариев написанных на одном из языков программирования, например **PHP**.



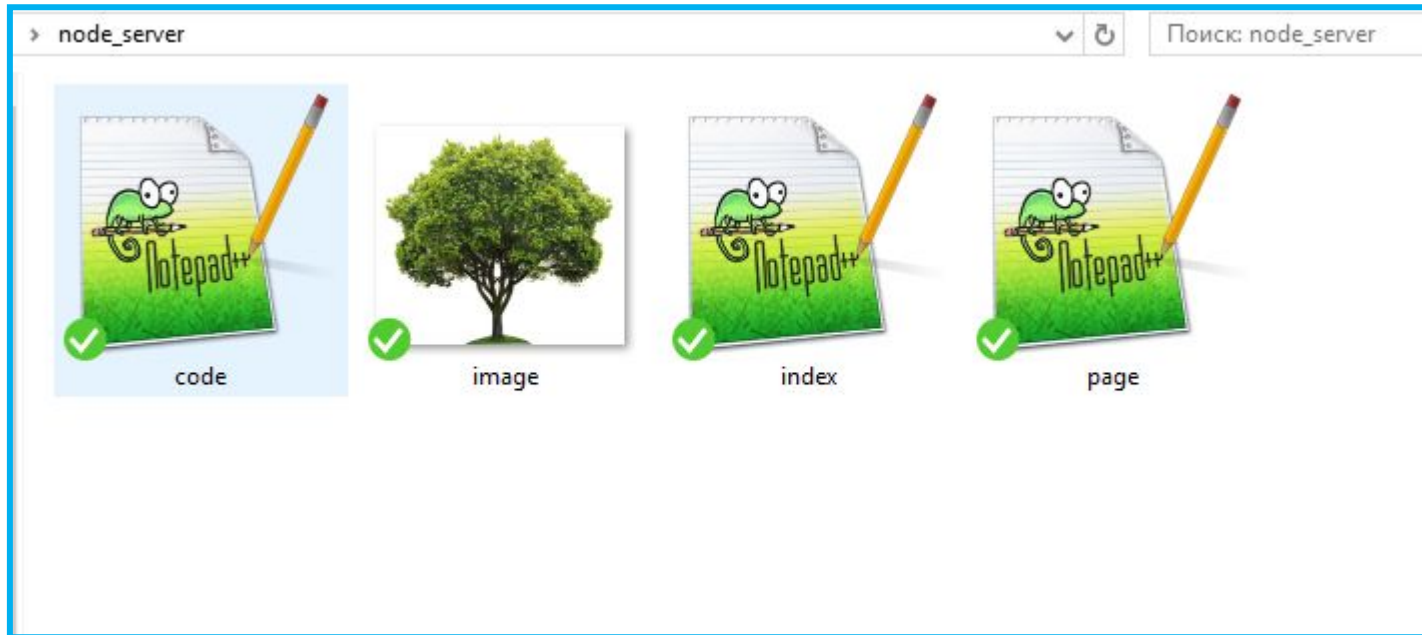
# NodeJS может быть HTTP-сервером (Web-сервером)



***NodeJS** может работать и HTTP-сервером, и заниматься обработкой данных на стороне сервера. И управляется всё это это языком **JavaScript**.*



***Небольшая заготовка для нашего сервера, скачайте и распакуйте его на диск.***



**[http://js.courses.dp.ua/files/node\\_server.zip](http://js.courses.dp.ua/files/node_server.zip)**

# HTTP сервер на базе NodeJS

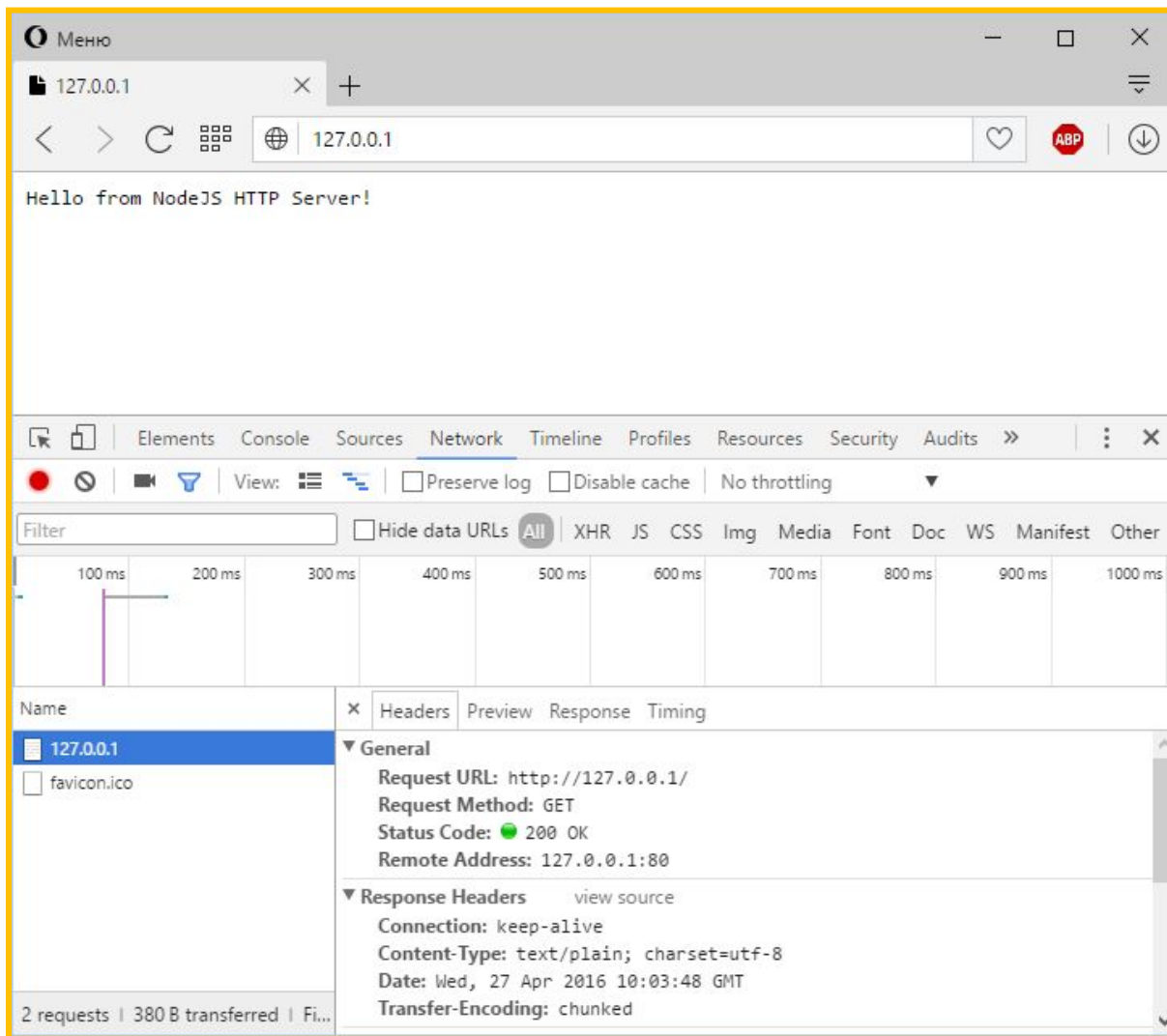
```
1  var http_module = require('http');
2
3  function onRequest(request, response) {
4      response.writeHead(200, { 'Content-Type': 'text/plain; charset=utf-8' });
5      response.end("Hello from NodeJS HTTP Server!");
6  }
7
8  var server = http_module.createServer(onRequest);
9  server.listen(80, '0.0.0.0');
```

*Сохраняем в **code.js** и запускаем **node code.js***

*Теперь сервер ждёт запросы от браузеров на все **ip-адреса** которые присвоены нашему компьютеру.*



# HTTP



*Если компьютер входит в сеть, то вы можете обратиться к запущенному серверу с других компьютеров.*

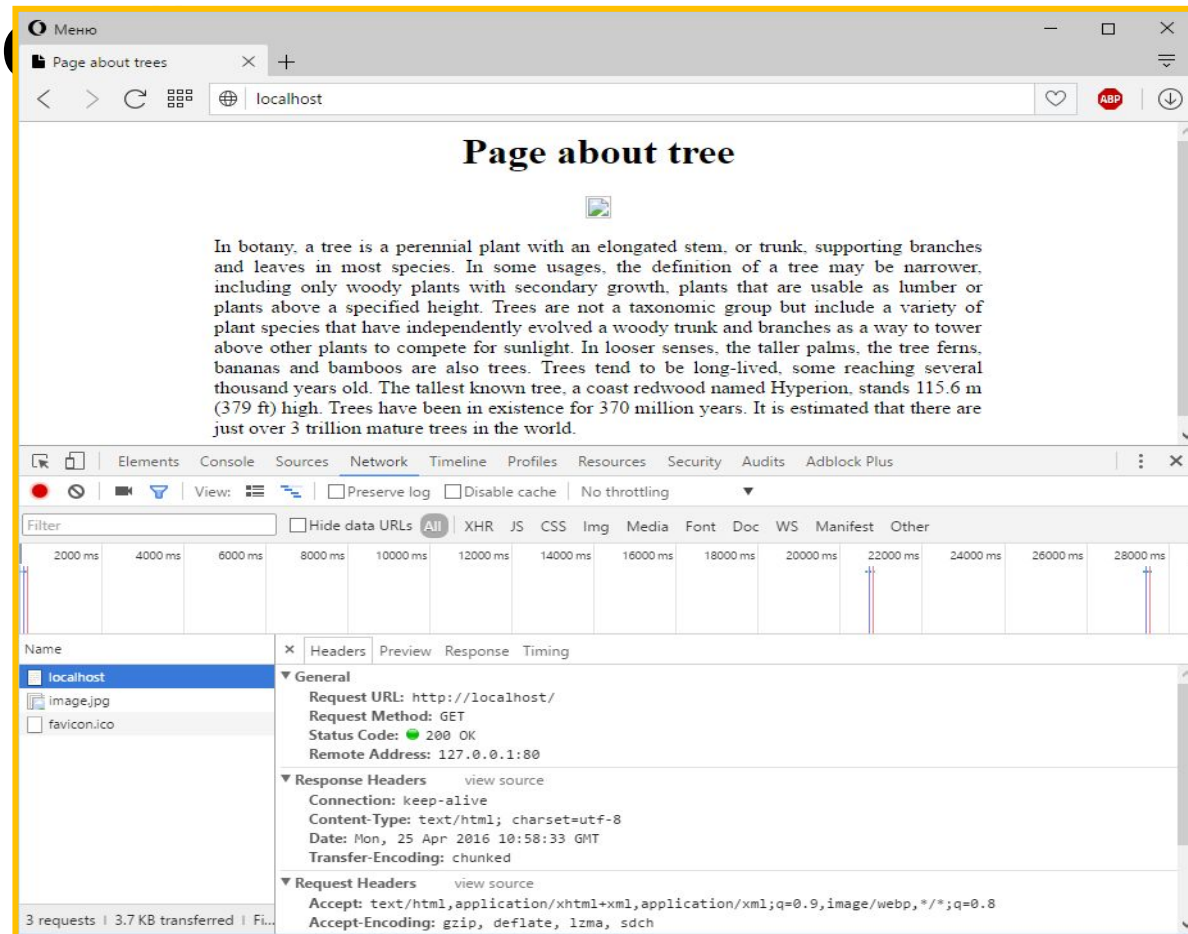
# HTTP сервер, возврат файла

```
1  var http_module = require('http');
2  var fs = require('fs');
3
4  function onRequest(request, response) {
5      fs.readFile('index.html', function(err, data) {
6          response.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });
7          response.write(data);
8          response.end();
9      });
10 }
11
12 var server = http_module.createServer(onRequest);
13 server.listen(80, '0.0.0.0');
```

*HTTP сервер ждёт запросов и на любой запрос в ответ возвращает HTML-файл (index.html) который лежит в том же каталоге что и файл с кодом.*



# HTTP сервер, возврат



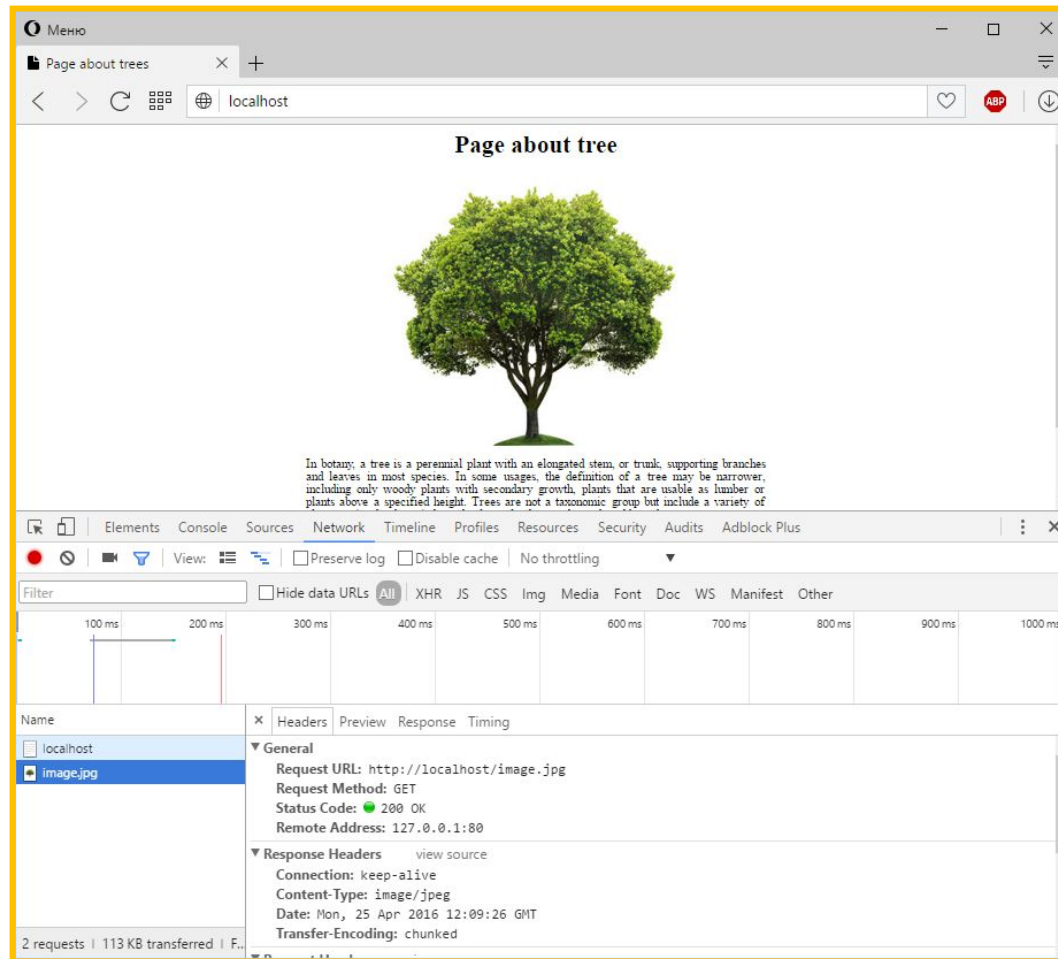
*HTTP сервер ждёт запросов и на любой запрос в ответ возвращает HTML-файл (index.html) который лежит в том же каталоге что и файл с кодом.*

# Статический HTTP

```
code.js
1  var http_module = require('http');
2  var file_system = require('fs');
3  var mime_types = require('mime');
4
5  function onRequest(request, response){
6
7      console.log("Request for: " + request.url);
8
9      var file_path = (request.url == "/" ? "./index.html" : "." + request.url);
10
11     file_system.exists(file_path, function(exists_result){
12         if(exists_result){
13             file_system.readFile(file_path, function(err, file_content) {
14                 response.writeHead(200, { 'Content-Type' : mime_types.lookup(file_path) });
15                 response.end(file_content);
16             });
17         }else{
18             response.writeHead(404, { 'Content-Type': 'text/plain' });
19             response.write("Error 404. Page not found.");
20             response.end();
21         }
22     });
23 }
24
25 var server = http_module.createServer(onrequest);
26 server.listen(80, '0.0.0.0');
```

*Статический HTTP-сервер просто возвращает запрашиваемый файл, или информацию о том, что файл не найден.*

# Статический HTTP



*Статический HTTP-сервер просто возвращает запрашиваемый файл, или информацию о том, что файл не найден.*

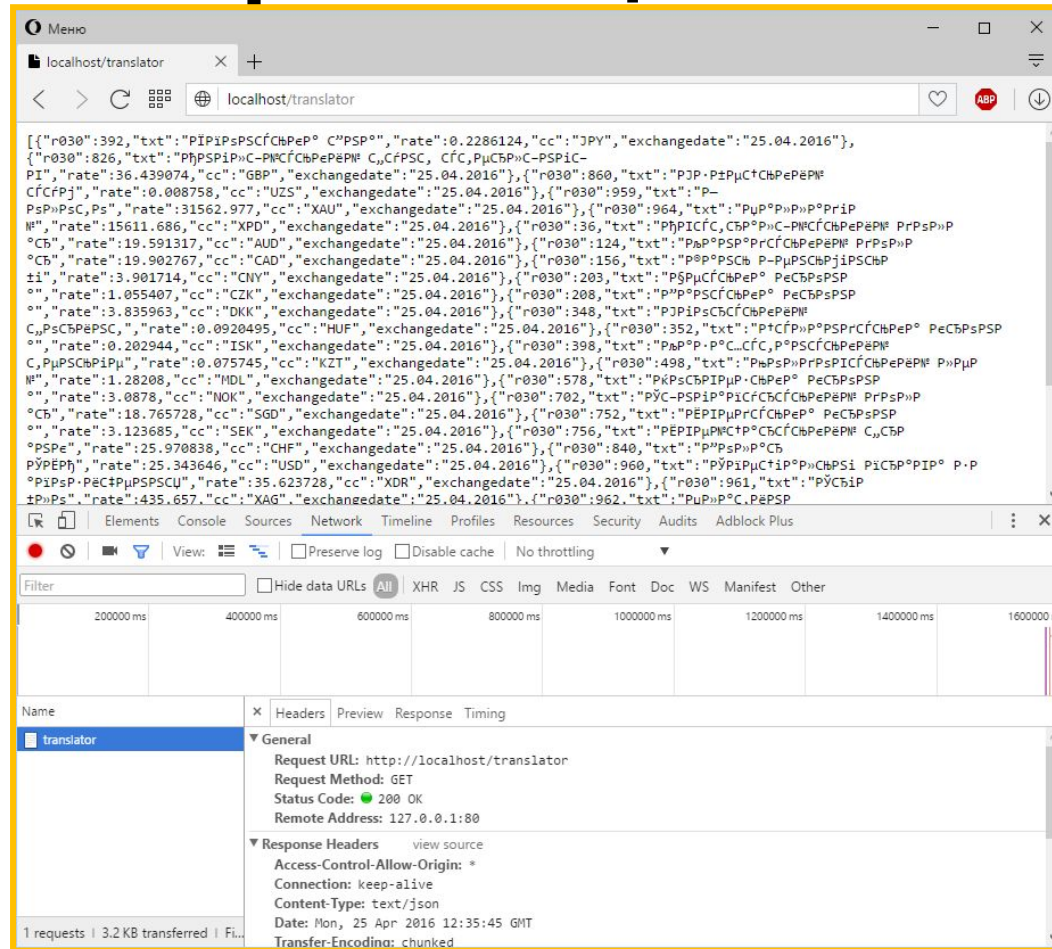
# Ретрансляция

```
code.js
1 var http_module = require('http');
2 var file_system = require('fs');
3 var mime_types = require('mime');
4 var request_module = require('request');
5
6 function onrequest(request, response){
7
8     console.log("Request for: " + request.url);
9
10    if(request.url == "/translator"){
11        request_module('http://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json', function(err, data, content){
12            response.writeHead(200, { 'Content-Type' : 'text/json', 'Access-Control-Allow-Origin': '*' });
13            response.end(content);
14        });
15        return;
16    }
17
18    var file_path = (request.url == "/" ) ? "./index.html" : ("." + request.url);
19
20    file_system.exists(file_path, function(exists_result){
21        if(exists_result){
22            file_system.readFile(file_path, function(err, file_content) {
23                response.writeHead(200, { 'Content-Type' : mime_types.lookup(file_path) });
24                response.end(file_content);
25            });
26        }else{
27            response.writeHead(404, { 'Content-Type': 'text/plain' });
28            response.write("Error 404. Page not found.");
29            response.end();
30        }
31    });
32 }
33
34 var server = http_module.createServer(onrequest);
35 server.listen(80, '0.0.0.0');
```

*На стороне сервера нет никаких ограничений для загрузки контента с других сайтов, поэтому наш сервер может выступать ретранслятором для загрузки данных со сторонних ресурсов.*



# Ретрансляция



На стороне сервера нет никаких ограничений для загрузки контента с других сайтов, поэтому наш сервер может выступать ретранслятором для загрузки данных со сторонних ресурсов.

# Node.js для

## *Node.js для начинающих*

Автор: [Manuel Kiessling](#)

Перевод: [Artod](#)

Правки: spmbt

### **О проекте**

Цель данного документа — помочь вам начать разработку приложений на Node.js и научить всему, что необходимо знать о «продвинутом» JavaScript. Это больше, чем обычный «Hello world»-тutorial.

### **Статус**

Вы читаете финальную версию этой книги, в обновлениях исправляются только ошибки или отражаются изменения в новых версиях Node.js.

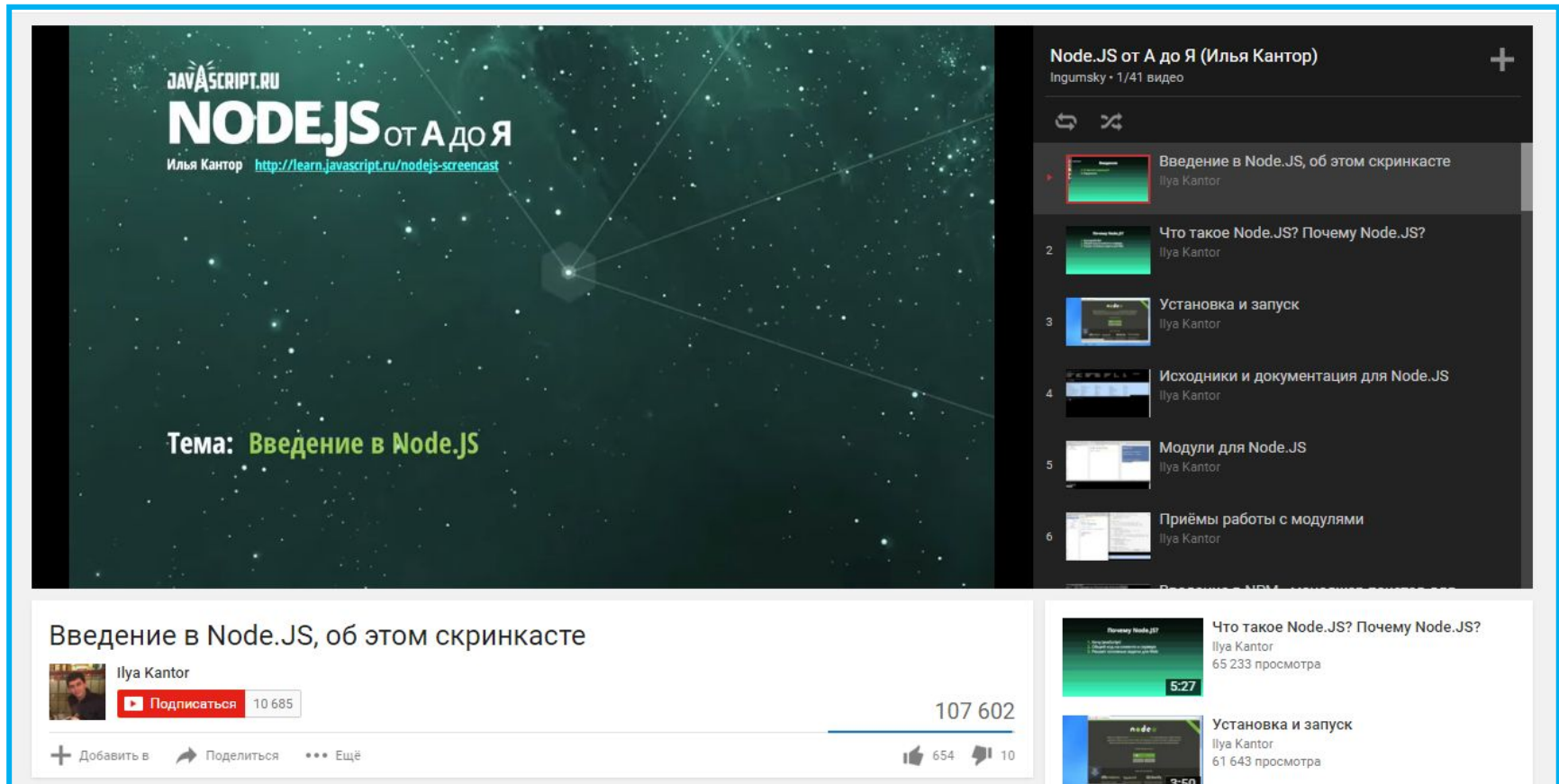
### **Целевая аудитория**

Вероятно, документ будет полезен читателям с базовыми знаниями,

<http://www.nodebeginner.ru/>

# Введение в Node.JS от Ильи Кантора\*

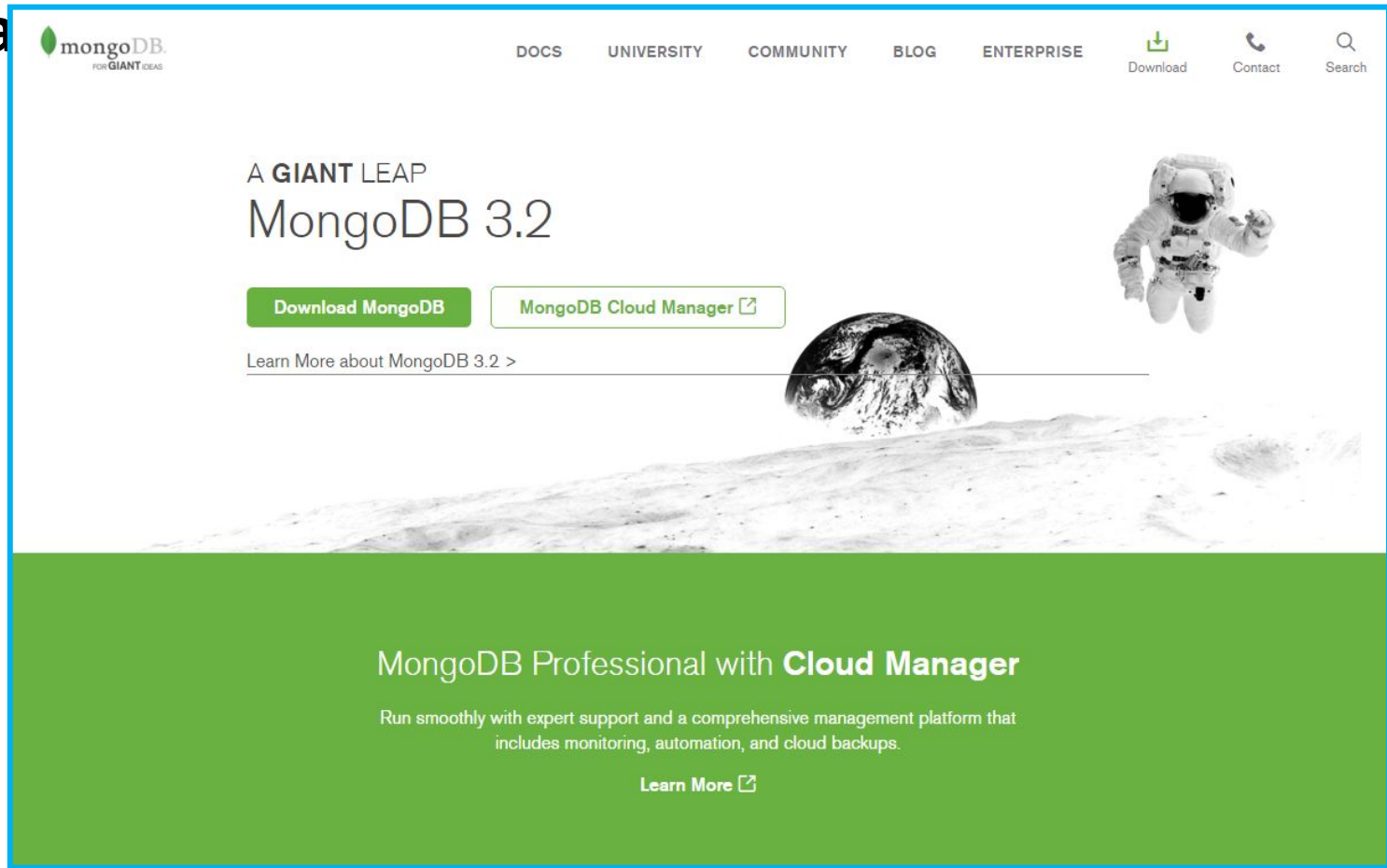
*\*создателя  
javascript.ru*



The screenshot shows a YouTube video player interface. The main video area displays a dark green background with a constellation pattern and the text "JAVASCRIPT.RU NODE.JS ОТ А ДО Я" and "Илья Кантор <http://learn.javascript.ru/nodejs-screencast>". Below this, it says "Тема: Введение в Node.JS". The video title is "Введение в Node.JS, об этом скринкасте" by Ilya Kantor, with 10 685 subscribers and 107 602 views. The video progress bar is at 0:00. To the right, a playlist is visible with six items: "Введение в Node.JS, об этом скринкасте", "Что такое Node.JS? Почему Node.JS?", "Установка и запуск", "Исходники и документация для Node.JS", "Модули для Node.JS", and "Приёмы работы с модулями". Below the playlist, two video thumbnails are shown: "Что такое Node.JS? Почему Node.JS?" (5:27) and "Установка и запуск" (3:50).

<https://www.youtube.com/watch?v=ILpS4Fq3lmw&list=PLsuEohlthXdkRSxJTkmTstWKHgBHsd3Dx>

# MongoDB – база данных под управлением Java



<https://www.mongodb.org/>

<http://metanit.com/nosql/mongodb/>



# Node.js для

## *Node.js для начинающих*

Автор: [Manuel Kiessling](#)

Перевод: [Artod](#)

Правки: spmbt

### **О проекте**

Цель данного документа — помочь вам начать разработку приложений на Node.js и научить всему, что необходимо знать о «продвинутом» JavaScript. Это больше, чем обычный «Hello world»-туториал.

### **Статус**

Вы читаете финальную версию этой книги, в обновлениях исправляются только ошибки или отражаются изменения в новых версиях Node.js.

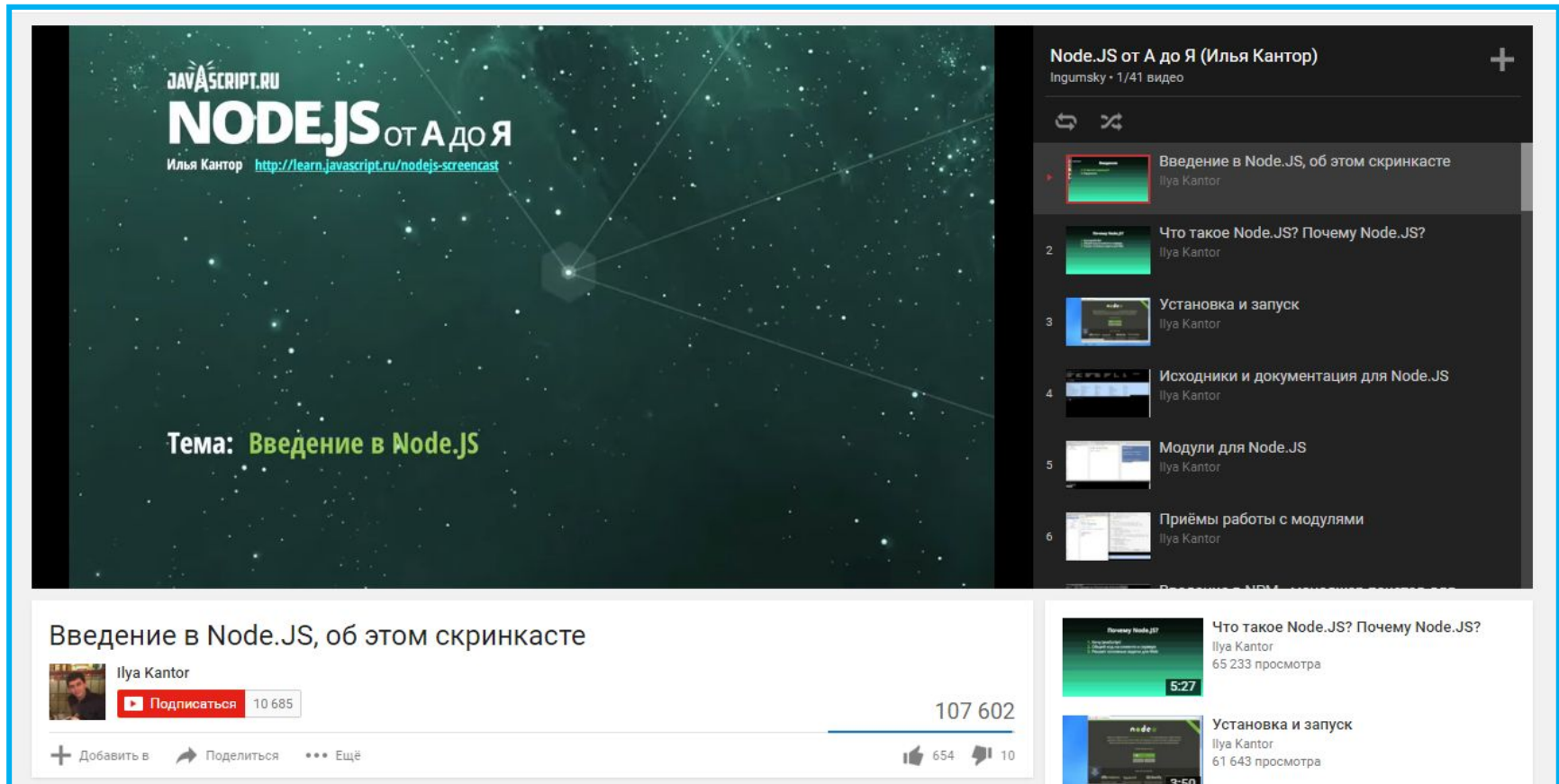
### **Целевая аудитория**

Вероятно, документ будет полезен читателям с базовыми знаниями,

<http://www.nodebeginner.ru/>

# Введение в Node.JS от Ильи Кантора\*

*\*создателя  
javascript.ru*



The screenshot shows a YouTube video player interface. The main video area displays a dark green background with a constellation pattern and the text "JAVASCRIPT.RU NODE.JS ОТ А ДО Я" and "Илья Кантор <http://learn.javascript.ru/nodejs-screencast>". Below this, it says "Тема: Введение в Node.JS". The video title is "Введение в Node.JS, об этом скринкасте" by Ilya Kantor, with 10 685 subscribers and 107 602 views. The video progress bar is at 0:00. To the right, a playlist of six videos is shown, including "Что такое Node.JS? Почему Node.JS?", "Установка и запуск", "Исходники и документация для Node.JS", "Модули для Node.JS", and "Приёмы работы с модулями".

Node.JS от А до Я (Илья Кантор)  
Ingumsky • 1/41 видео

Введение в Node.JS, об этом скринкасте  
Илья Кантор

2 Что такое Node.JS? Почему Node.JS?  
Илья Кантор

3 Установка и запуск  
Илья Кантор

4 Исходники и документация для Node.JS  
Илья Кантор

5 Модули для Node.JS  
Илья Кантор

6 Приёмы работы с модулями  
Илья Кантор

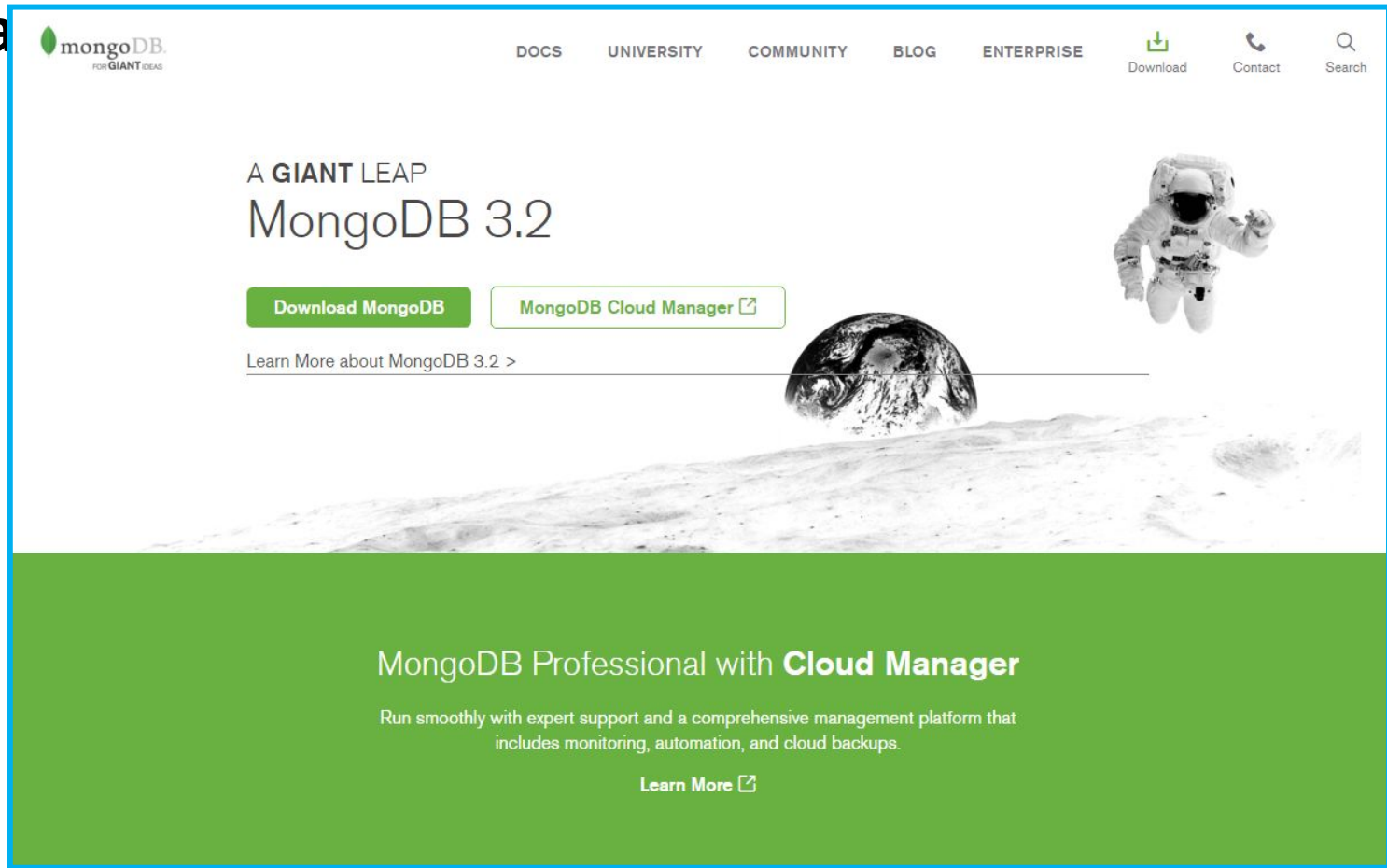
Введение в Node.JS, об этом скринкасте  
Илья Кантор  
Подписаться 10 685  
107 602  
+ Добавить в Поделиться Ещё  
654 10

Что такое Node.JS? Почему Node.JS?  
Илья Кантор  
65 233 просмотра  
5:27

Установка и запуск  
Илья Кантор  
61 643 просмотра  
3:50

<https://www.youtube.com/watch?v=ILpS4Fq3Imw&list=PLsuEohlthXdkRSxJTkmTstWKHgBHsd3Dx>

# MongoDB – база данных под управлением Java

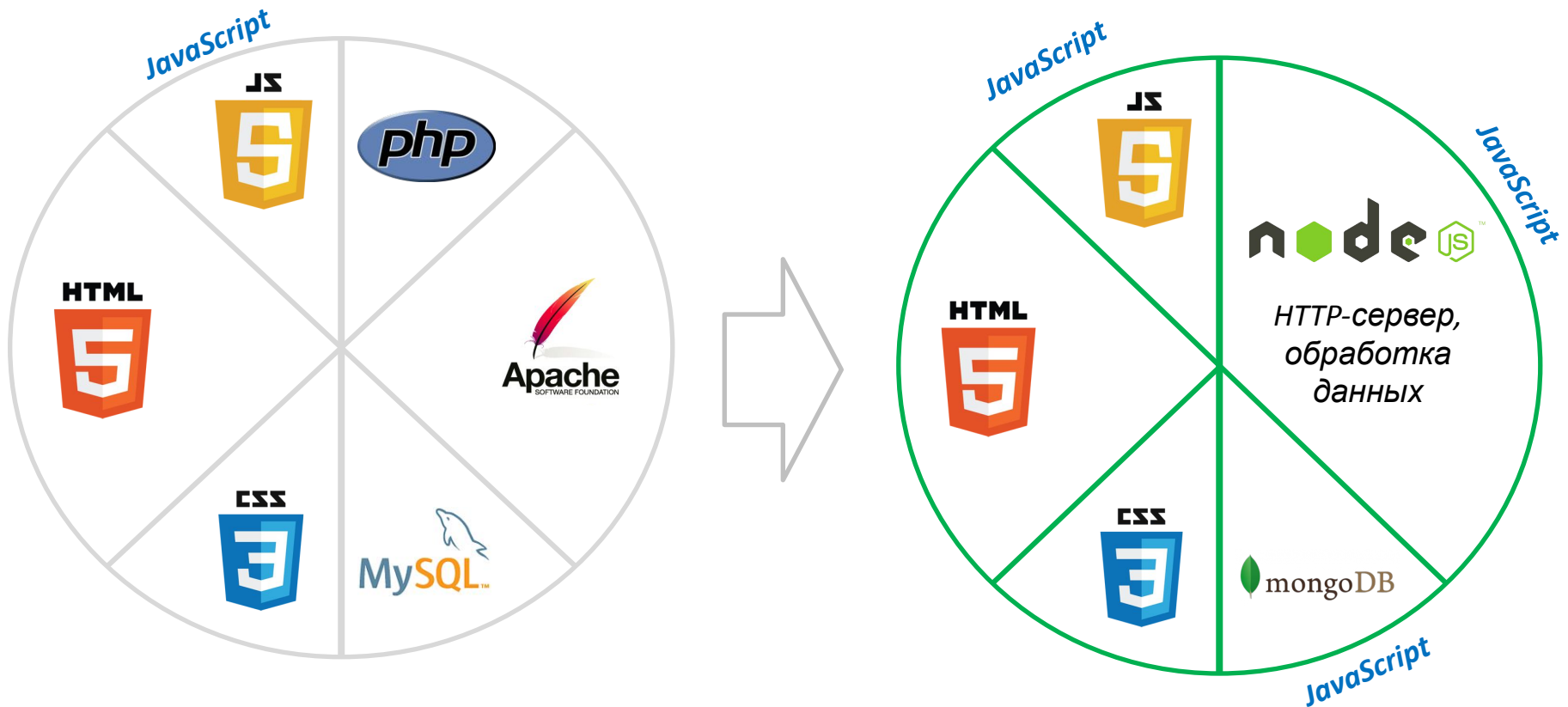


<https://www.mongodb.org/>

<http://metanit.com/nosql/mongodb/>

# Подведем итоги

*JavaScript – обычный язык программирования, с необычной сферой применения.*



*Технологии применяемый в веб-разработке.*

# Что мы учили и зачем?

## БЛОК «ОСНОВЫ ПРОГРАММИРОВАНИЯ»

- ***Переменные и типы данных;***
- ***Условные операторы  
(ветвление);***
- ***Циклы;***
- ***Структуры данных (массивы);***
- ***Функции;***
- ***Объекты.***

*Концепции которые свойственны подавляющему  
количеству языков программирования.*

# Что мы учили и зачем?

## БЛОК «ИНФРАСТРУКТУРА БРАУЗЕРА»

- *Роль JS в веб-разработке (фронт-енд);*
- *Инфраструктура браузера;*
- *Объектная модель документа (DOM);*

- **JavaScript как средство**  
*Браузер та самая виртуальная машина и интерпретатор кода, который понимает и выполняет код на JavaScript.*  
**манипуляции HTML-документом.**

# Что мы учили и зачем?

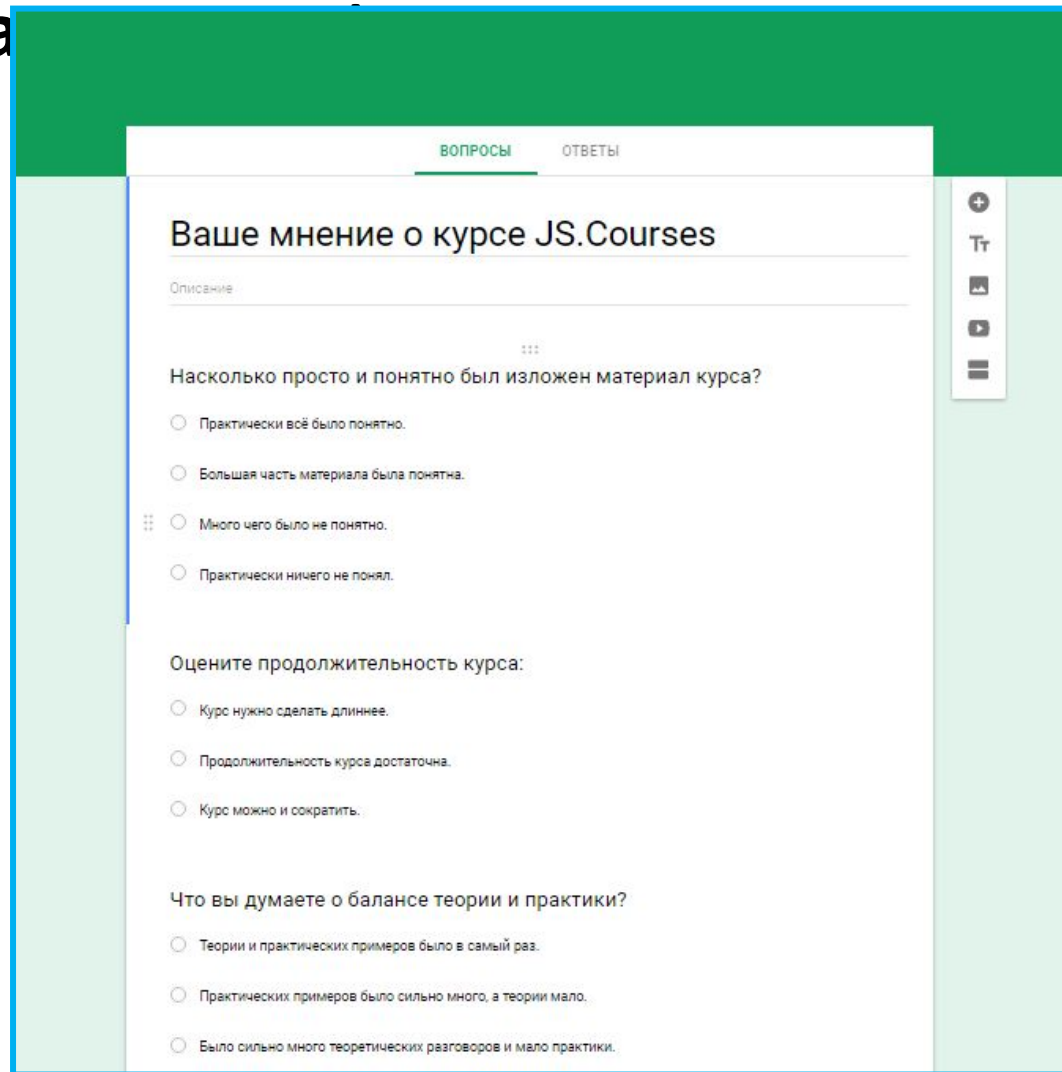
БЛОК  
«ПРАКТИКИ»

- ***Валидация и обработка данных;***
- ***Асинхронное взаимодействие с сервером: AJAX, Web-API, JSON;***
- ***JavaScript библиотеки и фреймворки: jQuery, AngularJS;***
- ***JavaScript вне браузера: NodeJS.***

*JavaScript прошел долгий путь развития от «игрушечного» языка до инструмента создания полноценного программного обеспечения.*

# Просьбы оставить отзыв

(а

A screenshot of a web form titled "Ваше мнение о курсе JS.Courses" (Your opinion on the JS.Courses course). The form is set against a green header and a light green sidebar. At the top, there are two tabs: "ВОПРОСЫ" (Questions) and "ОТВЕТЫ" (Answers). Below the title is a text input field labeled "Описание" (Description). The main content area contains three sections of questions, each with four radio button options. The first section asks "Насколько просто и понятно был изложен материал курса?" (How simple and understandable was the course material presented?). The second section asks "Оцените продолжительность курса:" (Evaluate the duration of the course:). The third section asks "Что вы думаете о балансе теории и практики?" (What do you think about the balance of theory and practice?). The sidebar on the right contains a vertical menu with icons for adding content, text, images, video, and a list icon.

ВОПРОСЫ    ОТВЕТЫ

## Ваше мнение о курсе JS.Courses

Описание

Насколько просто и понятно был изложен материал курса?

- ☐ Практически всё было понятно.
- ☐ Большая часть материала была понятна.
- ☐ Много чего было не понятно.
- ☐ Практически ничего не понял.

Оцените продолжительность курса:

- ☐ Курс нужно сделать длиннее.
- ☐ Продолжительность курсов достаточна.
- ☐ Курс можно и сократить.

Что вы думаете о балансе теории и практики?

- ☐ Теории и практических примеров было в самый раз.
- ☐ Практических примеров было сильно много, а теории мало.
- ☐ Было сильно много теоретических разговоров и мало практики.

<http://js.courses.dp.ua/feedback>