

Синтаксис

Запуск консоли (REPL)	node
Запустить сервер	node файл.js
остановить сервер	ctrl+c
Подключить пакет	var something = require('пакет');
Выполнить функцию подключенного пакета	something();

npm

Установка пакета

```
npm install пакет
```

Установка с сохранением в dependencies (package.json)

```
npm install пакет --save
```

ejs

Установка ejs

```
npm install ejs --save
```

Синтаксис

Весь код пишется как html, куски js-кода вставляют в конструкции: `<%= код %>` - когда получаемое значение выводится в html (например вывод переменных) `<% код %>` - когда код в конструкции не выводится (условия, циклы и т.п.)

Условие

```
<% if(условие) { %>
  <p>Выведется, только если
  выполняется условие</p>
<% } %>
```

ejs (cont)

Цикл

```
<% for(var i=0;i<posts.length;i++) { %>
  <div><%= posts[i].content %></div>
<% } %>
```

Передать переменные в ejs-файл

```
res.render("файл.ejs", {переменная: "значение"}); - второй параметр - объект
```

Шаблоны (partials)

Хранятся в папке "views/partials"
Пример шаблона header.ejs:
`<!DOCTYPE html>...<body>`

Включить шаблон

```
<% include partials/header %>
```

package.json

package.json хранит метаданные проекта

В разделе 'dependencies' хранятся используемые модули и версии

Сгенерировать package.json: `npm init`
entry point - начальный файл (обычно app.js)
пропустить пункт - нажать enter

Express

Подключение express

```
var express = require('express');
var app = express(); - сохр. в переменную app
```

Routes

обрабатывают http-запросы
Пример:
`app.get("/", function(req, res) { res.send("text"); });`
req, res - передаем объекты запроса и ответа в функцию

Express (cont)

Ожидать запросы

```
app.listen(порт, process.env.IP, callback);
Порт обычно 3000 или process.env.PORT
```

Остальные запросы (404)

ссылка "*"
Так как выполняется первый подходящий route, то route для 404 должен идти последним

Редирект

```
res.redirect("/путь");
```

Параметры (переменные)

```
app.get("/:post/id/:title", function);
:post и :title - параметры
```

Доступ к параметру

```
var параметр = res.params.параметр;
Параметры лучше обрабатывать через toLowerCase()
```

Параметр в число

```
Number(req.params.параметр);
```

Ответ

```
.send() - ответ, выполняется один раз
.render("файл") - ответ файлом (ejs/html)
```

Расширение файлов для render

```
app.set("view engine", "ejs"); - в функции render можно будет указывать только имя файла, без расширения (ejs)
```

Express (cont)

Стили и скрипты

css и **js** файлы хранить в папке public (можно в подпапках)
Папку нужно подключить в app.js:
`app.use(express.static("public"))`; - в адресах
файлов не нужно будет указывать public ("/js/main.js",
"css/app.css")

POST

```
app.post("/url", function);
```

Получить атрибут name тега input (POST)

```
req.body.input_name; - для доступа к body нужен пакет  
body-parser
```

Подключение body-parser

```
var bodyParser = require("body-parser");  
app.use(bodyParser.urlencoded({extended:  
trued}));
```

C

By **Yur375**

cheatography.com/yur375/

Not published yet.

Last updated 24th October, 2018.

Page 2 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>