

# Консольные программы на Node.js

# Содержание урока

- Параметры и окружение
- Работа с консолью
- Работа с файлами
- Обработка ошибок
- Полезные библиотеки

# Параметры и окружение

- `process`
- `process.argv`
- Пример: `console.dir(process.argv);`
- `util.inspect(object, {depth: null});`
- <https://github.com/substack/minimist>
- `process.env`
- `env.js`

# ВВЫВОД В КОНСОЛЬ

- `console.log`
- `console.error`
- `console.dir`
- `util.inspect`

# ВВОД ИЗ КОНСОЛИ

```
var readline = require('readline');

var rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

rl.question("What do you think of node.js? ", function(answer) {
  // TODO: Log the answer in a database
  console.log("Thank you for your valuable feedback:", answer);
  rl.close();
});
```

<https://github.com/flatiron/prompt>

<https://github.com/chriso/cli>

# Работа с файлами

- `fs.readFile(filename, [options], callback)`
- `fs.writeFile(filename, data, [options], callback)`
- `fs.appendFile(filename, data, [options], callback)`
- `*Sync`

# Работа с файлами

```
// Загрузка модуля fs (файловая система)
var fs = require('fs');

// Считывание содержимого файла в память
fs.readFile('example_log.txt', function (err, logData) {
    // Если произошла ошибка, то мы генерируем исключение,
    // и работа приложения завершается
    if (err) throw err;

    // logData имеет тип Buffer, переводим в string
    var text = logData.toString();
});
```

<https://github.com/nickewing/line-reader>

# Обработка ошибок

```
try {  
    throw new Error('my silly error');  
} catch (err) {  
    console.error(err);  
}
```

-----

```
process.on('uncaughtException', function(err) {  
    console.log('Caught exception: ' + err);  
});
```

-----

API Domains



# Полезные библиотеки

- underscore
- sugar
- async