



# Хроники серверного Жаваскрипта

Версия 2.0

**Марат Абдуллин**

Разработчик интерфейсов

Я.Субботник, Санкт-Петербург, 3 декабря 2011 года

# JavaScript ?????!!!??!1адин



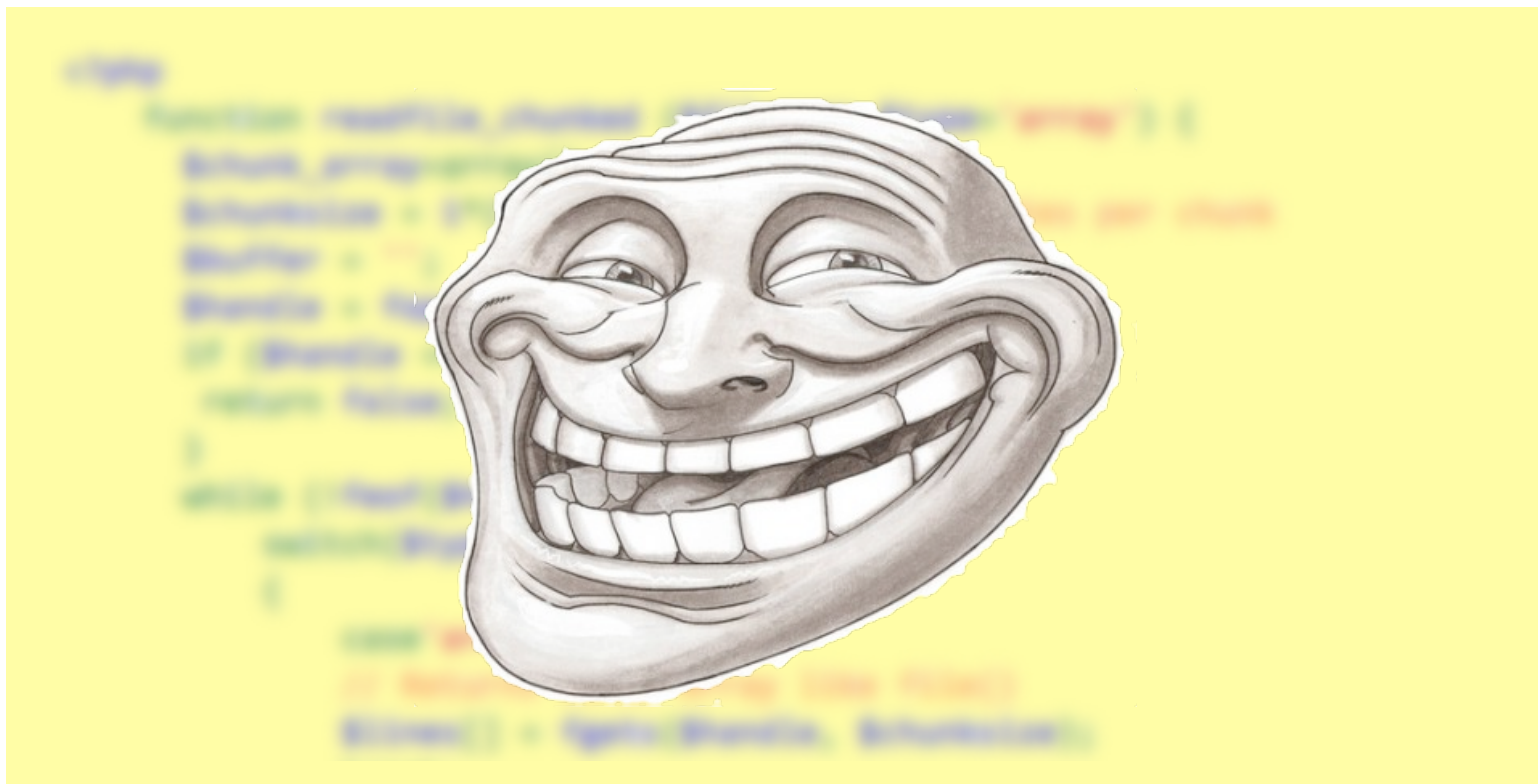
**Все еще используешь LAMP?!**

# **Прототипная парадигма объектно-ориентированного программирования!**

# Наш SERP\* написан на PHP4!

SERP — Search Engine Result Page

# Уникальная возможность Код СЕРПа!



**We put a function in your function, so you can call your function in a function call.**



**Марат Абдуллин**

Начинающий тролль-любитель

[hoho@yandex-team.ru](mailto:hoho@yandex-team.ru)



# Server-side JavaScript

Версия 3.0

**Марат Абдуллин**

Разработчик интерфейсов

Я.Субботник, Санкт-Петербург, 3 декабря 2011 года



# История JavaScript не стар, но и не юн

- Появился в 1995 году
- Стандартизован в 1997 году
- Активно развивается

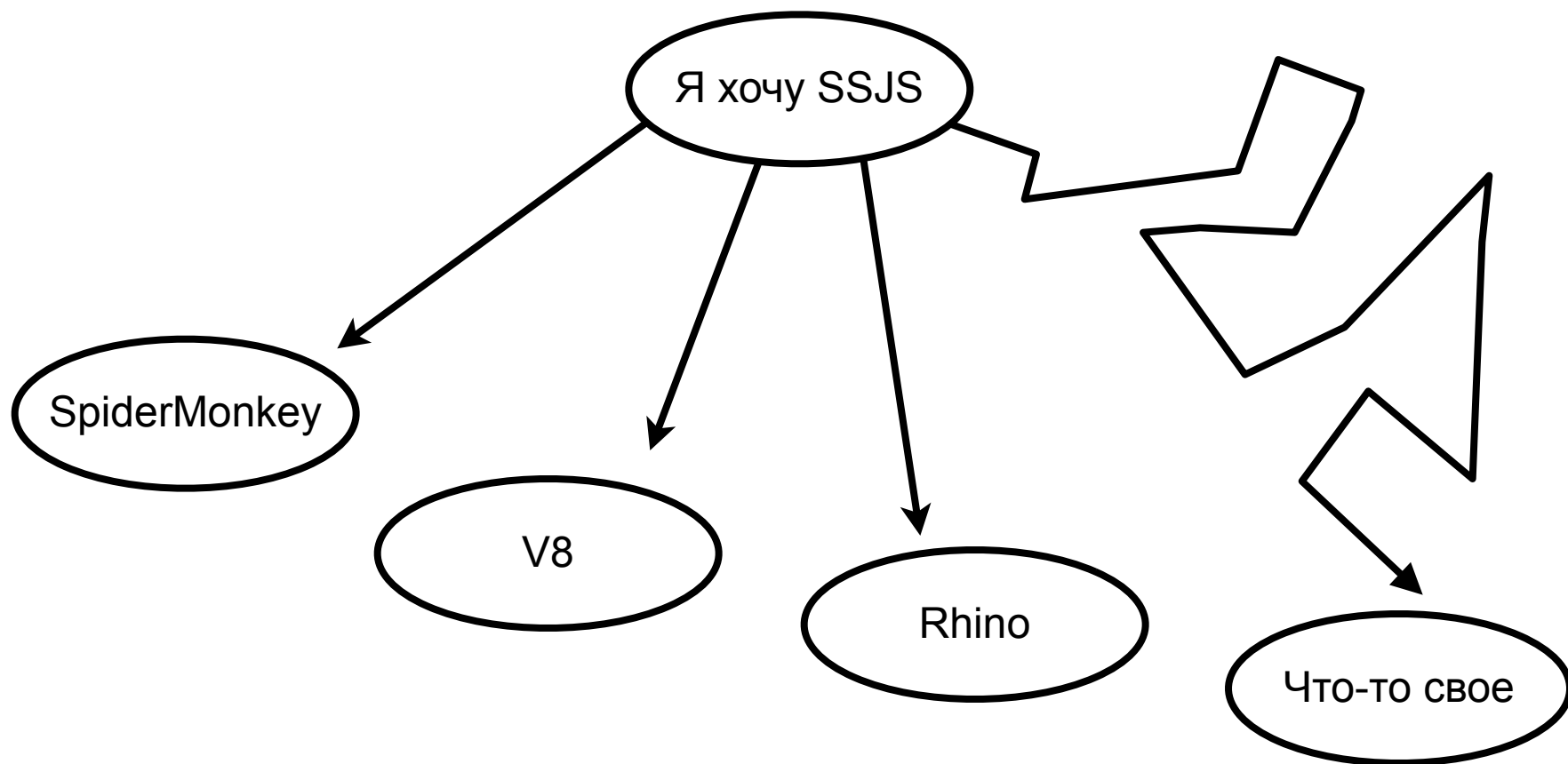


# История Server-side JavaScript есть давно

ESXX (R) Akshell (V8) Cocoon Flowscript (R)  
Juice (SM) Acre (R) OpenMocha (R) Myna (R) Webpop (R) RingoJS (R)  
DovetailDb (R) GLUEScript (SM) jslibs (SM) AppJet (R) im-jssp (R)  
Opera (\*) JSSP (R) ASP (\*) Xpages (\*) SSJS (SM) Node.js (V8) Jack (\*)  
firecat (R) PyV8 (V8) Wakanda (\*)  
Aptana Jaxer (SM) Torino (R) v8cgi (V8) Narwhal (R) Alfresco (R)  
SSX (R) wxJavaScript (SM) Rhino in Spring (R) jsdb (SM)  
Livelihood (SM) CouchDB (SM) Apache Sling (R) Whitebeam (SM) Phobos (R)  
10gen (R) Persevere (R) Rhinola (R) Junction (R) Google Apps Script (\*)  
Ejscript (\*) NOTEX (SM) POW (SM) Eclipse e4 (R) mod js (SM)  
jsext (SM) AppengineJS (R) JavaLikeScript (SM) APE (SM)  
GromJS (SM) Helma (R) Axiom Stack (R) EditMe (R) Erbix (R)



# Server-side JavaScript Engine Четыре варианта



# Server-side JavaScript Engine Embedding

```
#include <v8.h>
using namespace v8;
int main(int argc, char* argv[]) {
    HandleScope handle_scope;
    Persistent<Context> context = Context::New();
    Context::Scope context_scope(context);

    Handle<String> source = String::New("'Hi' + ', World!'");

    Handle<Script> script = Script::Compile(source);
    Handle<Value> result = script->Run();
    context.Dispose();
    String::AsciiValue ascii(result);
    printf("%s\n", *ascii);
    return 0;
}
```



# Server-side JavaScript Node.JS

- Модный асинхронный подход, хорошо сочетающийся с фронтэндами
- Очень производительный интерпретатор внутри
- Большая степень распространенности JavaScript-а
- Активно развивается



# Node.JS

## Пример модуля

```
#include <v8.h>
#include <node.h>
using namespace v8;
using namespace node;
Persistent<FunctionTemplate> constructor_template;
Handle<Value> Yo(const Arguments& args) {
    HandleScope scope;

    return String::New((char *) "Hi, World!");
}
void Init(Handle<Object> target) {
    HandleScope scope;
    Local<FunctionTemplate> t = FunctionTemplate::New(Yo);
    constructor_template = Persistent<FunctionTemplate>::New(t);
    t->InstanceTemplate()->SetInternalFieldCount(1);
    target->Set(String::NewSymbol("yo"), t->GetFunction());
}
extern "C" void
init (Handle<Object> target) {
    HandleScope scope;
    Init(target);
}
```

# А у нас в квартире — газ

## Что используем мы

- SpiderMonkey в Почте и в Xscript-e
- Несколько сервисов на Node.JS
- JavaScript-шаблонизаторы (xjst, yate)
- Утилиты сборки и не только (bem-tools)



**Марат Абдуллин**  
Разработчик интерфейсов

[hoho@yandex-team.ru](mailto:hoho@yandex-team.ru)