

СКРИПТОВЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

ВВЕДЕНИЕ

Лекция 1

Скрипт?

Сценарий – исполняемая процедура или набор исполняемых процедур.

Язык сценариев – высокоуровневый язык программирования для написания сценариев – кратких описаний действий, выполняемых системой.

Динамический язык – язык программирования, который позволяет определять типы данных, осуществлять синтаксический анализ и компиляцию «на лету», на этапе выполнения программы. Удобен для быстрой разработки приложений.

К динамическим языкам относятся: [Perl](#)К динамическим языкам относятся: Perl, [Tcl](#)К динамическим языкам относятся: Perl, Tcl, [Python](#)К динамическим языкам относятся: Perl, Tcl, Python, [PHP](#)К динамическим языкам относятся: Perl, Tcl, Python, PHP, [Ruby](#)К динамическим языкам относятся: Perl, Tcl, Python, PHP, Ruby, [Smalltalk](#)К динамическим языкам относятся: Perl, Tcl, Python, PHP, Ruby, Smalltalk, [JavaScript](#).

Классификация

По применению языки можно разделить:

- командно-сценарные;
- прикладные сценарные;
- языки разметки;
- универсальные сценарные.

Командно-сценарные языки

Появились в 1960-х годах для управления заданиями в операционных системах. Эти языки чаще всего используются в пакетном режиме обработки.

- [JCL](#) (Job Control Language)
- [Sh](#) (Bourne shell)
 - [Bash](#) (Bourne again shell)
 - [Csh](#) (C shell)
 - [Ksh](#) (Korn shell)
- Pilot
- [REXX](#) (REstructured eXtended eXecutor)
- [AppleScript](#) (для Mac OS X)
- [COMMAND.COM](#) COMMAND.COM и [cmd.exe](#)
- [VB Script](#)
- [PowerShell](#)

Прикладные сценарные языки

Сценарные языки этого типа начали появляться в 1980-е годы, когда на персональных компьютерах стало возможным интерактивное общение с ОС. В клиент-серверной архитектуре такие языки работали как клиент.

- [AutoLISP](#) (для AutoCAD)
- [Emacs Lisp](#)
- [ERM](#)
- [Game Maker Language](#)
- [MQL4](#) script
- [UnrealScript](#)
- [VBA](#)
- LotusScript

К этой категории можно отнести ECMAScript и его диалекты (JScript, [JavaScript](#)).

Языки разметки

Главная характерная черта этих языков — встраивание специальных кодов (тегов) в обычный текст не только для целей структурирования и форматирования, но и для определения динамического поведения.

- GML
- [TeX](#)
- [SGML](#)
- [XML](#) (*eXtensible Markup Language*)

Родственны этим языкам [PostScript](#) Родственны этим языкам PostScript и [RTF](#).

Универсальные языки

Этот тип сценарных языков наиболее известен, особенно в применении к программированию для веба. Языки этого типа стали возникать с 1990-х годов.

- [Tcl](#) (Tool command language)
- [Lua](#)
- [Perl](#)
- [PHP](#)
- [Python](#)
- [REBOL](#)
- [Ruby](#)

Следует заметить, что многие языки этой категории имеют более широкое применение, чем в качестве просто языков сценариев.

Место скриптовых языков

- Скриптовые языки позволяют разработчикам сцеплять вместе различные пакеты программ, и согласовывать полученные в результате системы.
- Используются в качестве полноценных базовых инструментальных платформ.

Например, многие крупные коммерческие Интернет-приложения сейчас программируются преимущественно на языках Perl, Python или PHP.

- Скриптовые языки используются для автоматизации задач системного администрирования.

Недостатки скриптовых языков

1. Время исполнения больше чем в компилируемых языках программирования.

Если скорость выполнения важна, то часть кода можно выполнить на языке низкого уровня (таком, как C или C++) и вставить такой код в скрипт.

2. Отсутствие хорошей интегрированной среды разработки (IDE).

Конечно, какие-то интегрированные среды разработки существуют, однако в них недостает мощности, как у Visual Studio.

3. Отсутствие маркетингового бюджета.

Многие динамические языки идеально подходят для многих проектов, однако им тяжело конкурировать с такими локомотивами маркетинга, как Sun (Java) и Microsoft (C#), которые продолжают продвигать свои технологии как единственно возможные.

Привлекательность скриптовых языков

1. Скриптовые языки обладают более сложным инструментарием и поддерживают более прогрессивные техники программирования.

Например, возможности сортировки данных в Perl встроены прямо в язык. Это означает что в язык встроены все основные инструменты программирования, что избавляет от необходимости создавать их самостоятельно, т.е. нужно писать меньше кода, что увеличивает производительность разработчика.

2. Скриптовые языки позволяют быстро выполнять доработку кода без потери времени на ожидание окончания компиляции.

3. Программированию на скриптовых языках проще научиться. Чтобы стать средним программистом на C++, необходим больший опыт работы, чем для того, чтобы стать средним программистом на PHP.

VBScript

Visual Basic Scripting Edition (VBScript) – это язык программирования от компании Microsoft, предназначен для создания сценариев (скриптов), является подмножеством языка **Visual Basic** и широко используется при создании **административных сценариев** в системе Windows.

VBScript поддерживается в **Windows Script Host (WSH)**, который устанавливается вместе с почти любой версией Windows.

Синтаксис VBScript является несколько **упрощенной версией** стандартного синтаксиса Visual Basic.

Например, в VBScript не поддерживается типизация: все переменные имеют тип Variant.

Сценарии на языке VBScript используются в следующих областях:

- автоматизация администрирования систем Windows;
- серверный программный код на страницах ASP в Web-приложениях;
- клиентские сценарии на Web-страницах.

JScript

JScript – это язык программирования от компании Microsoft, предназначен для создания сценариев и является реализацией стандарта ECMAScript.

Синтаксис JScript во многом аналогичен языку JavaScript от компании Netscape. JScript по умолчанию поддерживается в Windows Script Host (WSH), который в свою очередь по умолчанию устанавливается вместе с почти любой версией Windows.

Сценарии на языке JScript используются в следующих областях:

- клиентские сценарии на Web-страницах;
- автоматизация администрирования систем Windows;
- серверный программный код на страницах ASP в Web-приложениях;
- язык JScript получил дальнейшее развитие в виде языка JScript.NET, который ориентирован на работу в рамках платформы Microsoft ASP.NET.

Python

Python (питон) – интерпретируемый, объектно-ориентированный язык программирования высокого уровня. Поддерживает классы, модули (которые могут быть объединены в пакеты), обработку исключений, а также многопоточную обработку.

Python относится к классу языков с динамической типизацией, предоставляет программисту автоматическую «сборку мусора» и удобные высокоуровневые структуры данных, такие как словари, списки, кортежи и др.

Интерпретатор языка Python распространяется свободно на основании лицензии Python Software Foundation (PSF) Licence, которая в некотором роде даже более демократична, чем GNU.

Официальный сайт проекта языка Python располагается по адресу <http://python.org/>. В стандартный комплект поставки Python входит интегрированная среда разработки IDLE.

Если не достаточно возможностей стандартной библиотеки, то существует множество библиотек, предоставляющих интерфейс ко всем мыслимым системным вызовам на разных платформах; в частности, на платформе Win32 поддерживаются все вызовы Win32 API, а также COM в объёме не меньшем, чем у Visual Basic или Delphi.

Количество прикладных библиотек для Python в самых разных областях без преувеличения огромно (веб, базы данных, обработка изображений, обработка текста, численные методы, приложения операционной системы, и т. д.).

Python легко расширяется языками C и C++, а на платформе Windows – также с помощью COM.

Библиотека Numeric Python для работы с многомерными массивами позволяет достичь производительности научных расчётов, сравнимой с MATLAB.

Существует специальная библиотека **psyco** (<http://psyco.sf.net/>), позволяющая оптимизировать выполнение некоторых программ, после чего скорость их выполнения можно сравнивать с программами на Си.

Tcl

Tcl (Tool Command Language) – интерпретируемый язык программирования высокого уровня.

Официальный сайт языка - <http://www.tcl.tk/>.

Tcl ориентирован преимущественно на автоматизацию рутинных процессов ОС и крупных программных систем, состоит из мощных команд, ориентированных на работу с абстрактными нетипизированными объектами.

Принципиальное отличие Tcl от командных языков ОС состоит в независимости от типа системы (когда не надо утруждать себя изучением нового командного языка) и, самое главное, он позволяет создавать переносимые программы с графическим интерфейсом (GUI).

Существует оболочка Visual Tcl, которая позволяет разрабатывать кроссплатформенное ПО для UNIX, Windows и Macintosh.

Ruby

Ruby – интерпретируемый скриптовый язык высокого уровня для быстрого и удобного объектно-ориентированного программирования.

Ruby имеет большое количество средств для обработки текстов, для решения системных задач. Ruby является полностью свободным языком программирования с возможностью копирования, модификации и распространения.

Ruby перенесён на множество платформ. Он разрабатывался на Linux, но работает на многих версиях Unix, DOS, Windows 95/98/Me/NT/2000/XP, Mac OS, BeOS, OS/2, и т.д. Целью создания Ruby был настоящий объектно-ориентированный интерпретируемый язык программирования.

Perl

Perl – интерпретируемый скриптовый язык программирования, один из самых распространённых в области веб-программирования.

По одной из версий, Perl — аббревиатура, которая расшифровывается как "Practical Extraction and Report Language" (практический язык извлечений и отчётов). Существует также ряд других вариантов. Согласно самому красивому из них, название "perl" произошло от слова "pearl" (жемчужина).

Основной особенностью языка считаются его богатые возможности для работы с текстом, реализованные при помощи регулярных выражений.

Перл также знаменит огромной коллекцией дополнительных модулей CPAN, находящейся по адресу <http://www.cpan.org/>.

PHP

PHP (пи-эйч-пи) – интерпретируемый скриптовый язык программирования, созданный для генерации HTML-страниц на веб-сервере и работы с базами данных.

В области веб-программирования PHP является на сегодняшний день одним из самых распространённых технологий (наряду с Perl, ASP/.NET и Python) благодаря простоте, скорости выполнения и богатой функциональности.

PHP распространяется свободно. Синтаксис языка похож на синтаксис C++. PHP поддерживается подавляющим большинством представителей сетевого хостинга.

PHP был создан в качестве надстройки над Perl для облегчения разработки веб-страниц. За свою жизнь PHP значительно изменялся.

Одной из сильнейших сторон PHP является возможность расширения ядра. Интерфейс написания расширений привлек к PHP множество сторонних разработчиков, что дало PHP возможность работать с огромным количеством баз данных, протоколов, поддерживать большое число API.

PHP поддерживает ООП (деструкторы, открытые, закрытые и защищённые члены и методы, final-члены и методы, интерфейсы и клонирование объектов). PHP поддерживает XML.

Плагин или скрипт?

Скрипт удобен

1. Программируемость без риска дестабилизировать систему, т.к. скрипты интерпретируются, а не компилируются, неправильно написанный скрипт выведет диагностическое сообщение, а не приведёт к системному краху.
2. Если важен выразительный код. Чем сложнее система, тем больше кода приходится писать «потому, что это нужно». А в скриптовом языке может быть совсем другая концепция программирования.
3. Скриптовый язык имеет собственный проблемно-ориентированный набор команд, и одна строка скрипта может делать то же, что несколько десятков строк на традиционном языке.
4. Как следствие, на скриптовом языке может писать программист очень низкой квалификации — например, [геймдизайнер](#) своими руками, не полагаясь на программистов, может корректировать правила игры.
5. Если требуется **кроссплатформенность**. Хорошим примером является [JavaScript](#) – его исполняют браузеры под самыми разными ОС.

Преимущества у плагинов

1. Готовые программы, оттранслированные в машинный код, выполняются значительно быстрее скриптов, которые **интерпретируются** из **исходного кода** динамически при каждом исполнении. Поэтому скриптовые языки не применяются для написания программ, требующих оптимальности и скорости исполнения.

Из-за простоты они часто применяются для написания небольших, однократных («проблемных») программ.

2. Полный доступ к любому аппаратному обеспечению или ресурсу ОС (в скриптовом языке для этого должен существовать написанный на машинном коде **API**). Плагины, работающие с аппаратным обеспечением, традиционно называют драйверами.

3. Если предполагается интенсивный обмен данными между основной программой и пользовательским расширением, то для плагина его обеспечить проще.