

Технологии разработки серверных приложений на РНР

Андрей Владимирович Попов

доцент кафедры анализа данных и ИИ

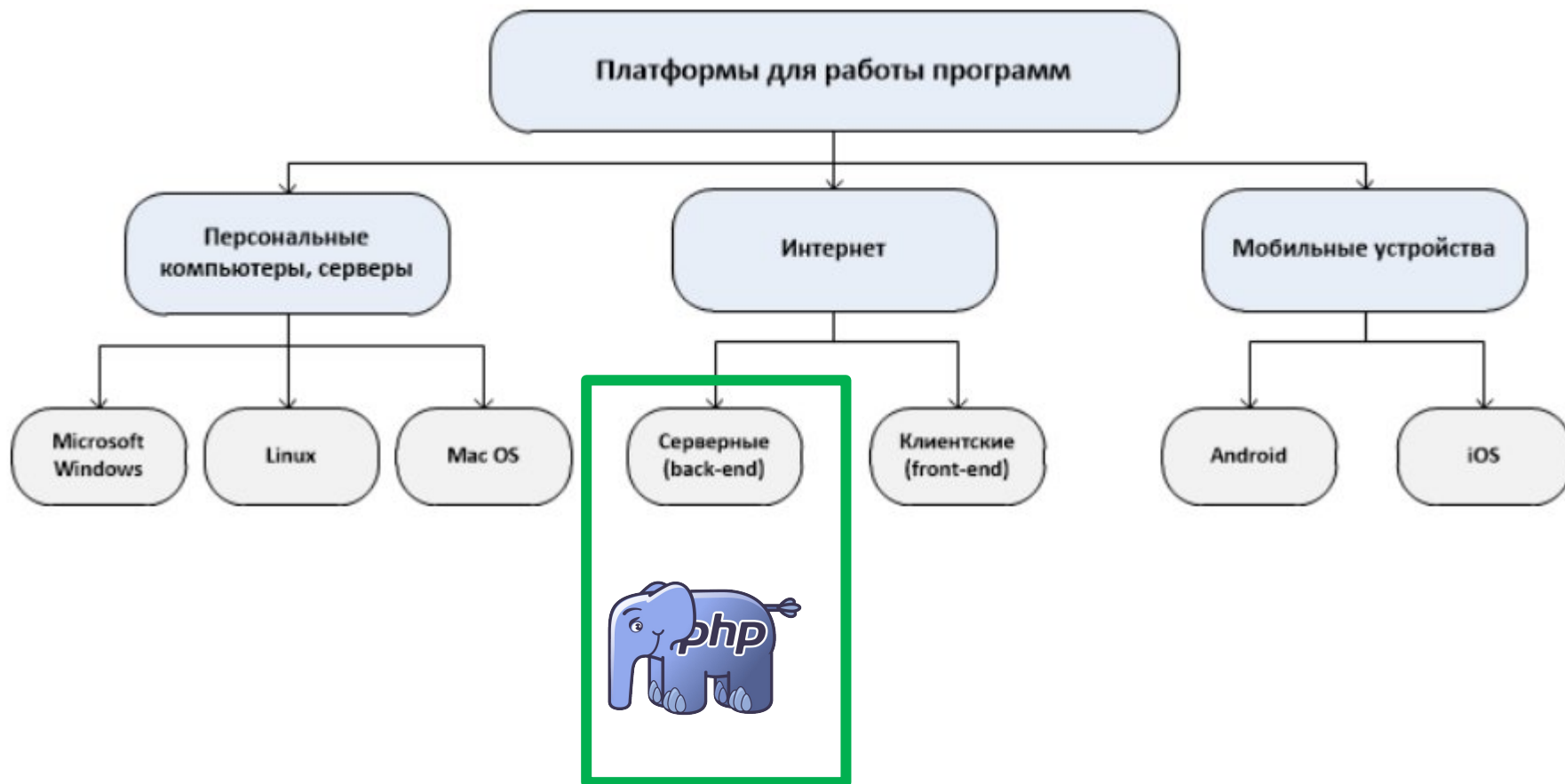
<https://andpop.ru>

Язык программирования и среда выполнения

Среда выполнения

Приложение, написанное
на языке
программирования

Язык программирования и среда выполнения



Веб-разработка на PHP





PHP, стандарт PSR,
PHP Standard Library

Git, Docker, make, Composer, PHPUnit,
PHPLint, PHP Code Sniffer, веб-
фреймворки, режимы выполнения

Синтаксис языка
программирования

Прикладные
библиотеки, фреймворки и
инструменты

Итераторы, генераторы

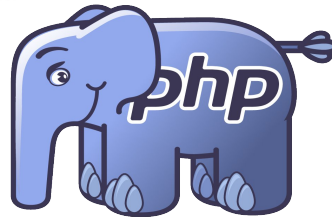
Алгоритмы и
структуры данных

Программирование

Практические приемы

Стек, очередь,
деревья

Архитектура (дизайн)
ПО



Групповая работа

Структурное процедурное программирование,
ООП, функциональные возможности, принципы
SOLID, паттерны проектирования

GitHub

Почему PHP?



Он же всё...

Или нет?

software system. The definition of the TIOBE index can be found [here](#).

Jan 2026	Jan 2025	Change	Programming Language	Ratings	Change
1	1		 Python	22.61%	-0.68%
2	4	▲	 C	10.99%	+2.13%
3	3		 Java	8.71%	-1.44%
4	2	▼	 C++	8.67%	-1.62%
5	5		 C#	7.39%	+2.94%
6	6		 JavaScript	3.03%	-1.17%
7	9	▲	 Visual Basic	2.41%	+0.04%
8	8		 SQL	2.27%	-0.14%
9	11	▲	 Delphi/Object Pascal	1.98%	+0.19%
10	18	▲▲	 R	1.82%	+0.81%
11	32	▲▲	 Perl	1.63%	+1.14%
12	10	▼	 Fortran	1.61%	-0.42%
13	14	▲	 Rust	1.51%	+0.34%
14	15	▲	 MATLAB	1.40%	+0.34%
15	13	▼	 PHP	1.38%	-0.00%
16	7	▼▼	 Go	1.24%	-1.37%

PHP ЖИВ?

PHP ЖИВ?

Язык	TIOBE	RedMonk	Stack Overflow	PYPL
Python	1	2	3	1
JavaScript	6	1	1	3
Java	3	3	7	2
C/C++	2	7	9	4
C#	5	5	8	5
PHP	13	4	11	7
Go	8	12	13	12
Rust	14	19	14	8
Ruby	21	9	19	17
Lua	33	-	16	18
Swift	26	11	20	11

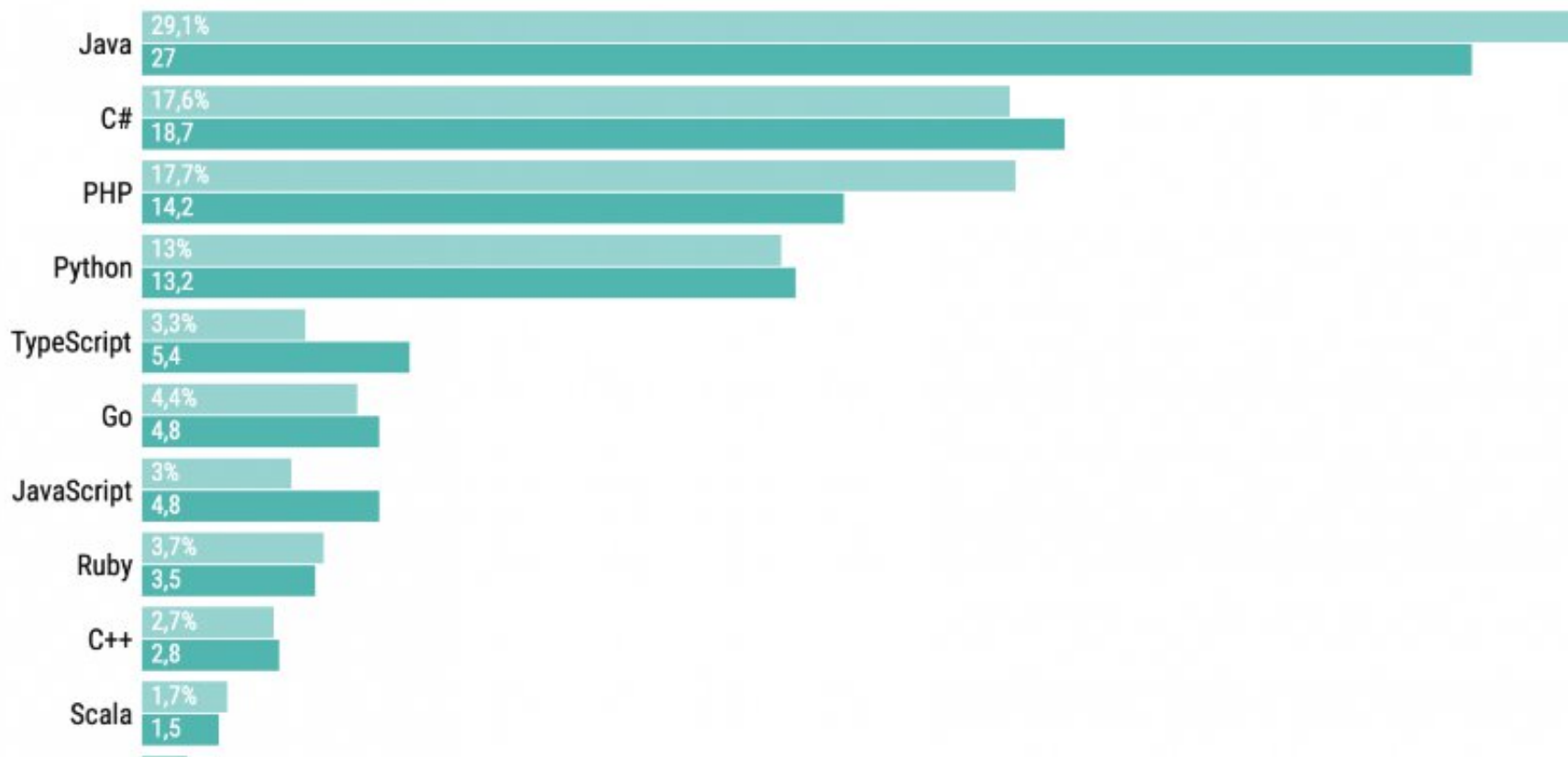
PHP ЖИВ?

Язык	Зарплата (₽)	Вакансий всего	Без опыта	Сложность
1C	72 000	151 028	1 261	Средняя
Python	100 000	11 331	827	Лёгкая
JavaScript	80 000	6 364	554	Лёгкая
Java	100 000	4 893	429	Средняя
C++ / C	100 000	3 523	311	Сложная
Swift	100 000	700	44	Средняя
C#	80 000	2 813	281	Средняя
PHP	80 000	2 908	206	Лёгкая
Lua	—	228	59	Средняя
Go	120 000	1 272	31	Сложная
Rust	100 000	167	2	Сложная
Ruby	85 000	266	0	Средняя

Рейтинг языков программирования по сферам использования

Back-end

2023 год

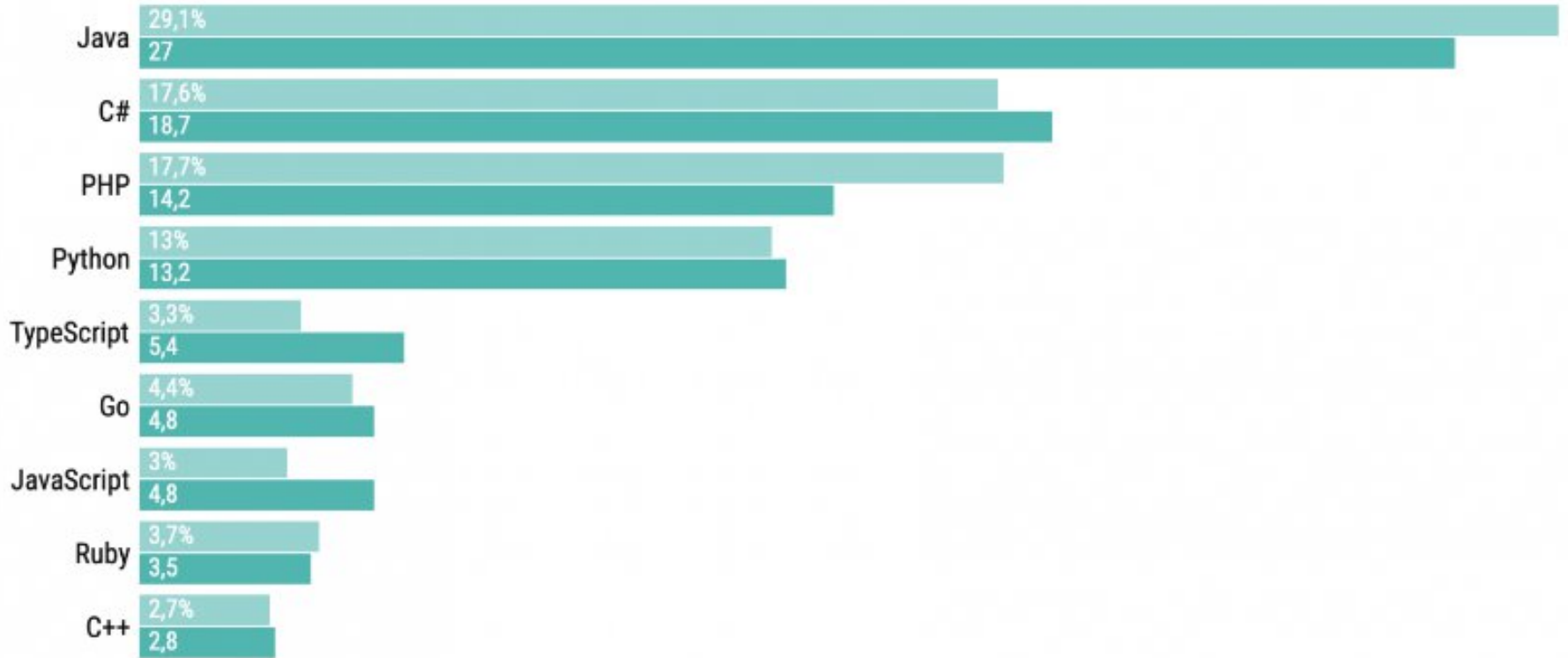


Почему PHP?

- Около 80% сайтов в Интернете написано на PHP
- Лендинги, блоги, соцсети, магазины и платформы, CMS (WordPress)
- Wikipedia, Facebook, Slack, Avito, Habr, VK, Blablacar, Яндекс Еда, Кинопоиск, SuperJob, Skyeng, Юла, Lamoda, Delivery Club, Перекрёсток, Wildberries, DNS, Tutu, Pikabu, Pornhub
- Низкий порог вхождения, много вакансий для начинающих
- PHP развивается, вбирая в себя лучшие решения и идеи из других языков программирования

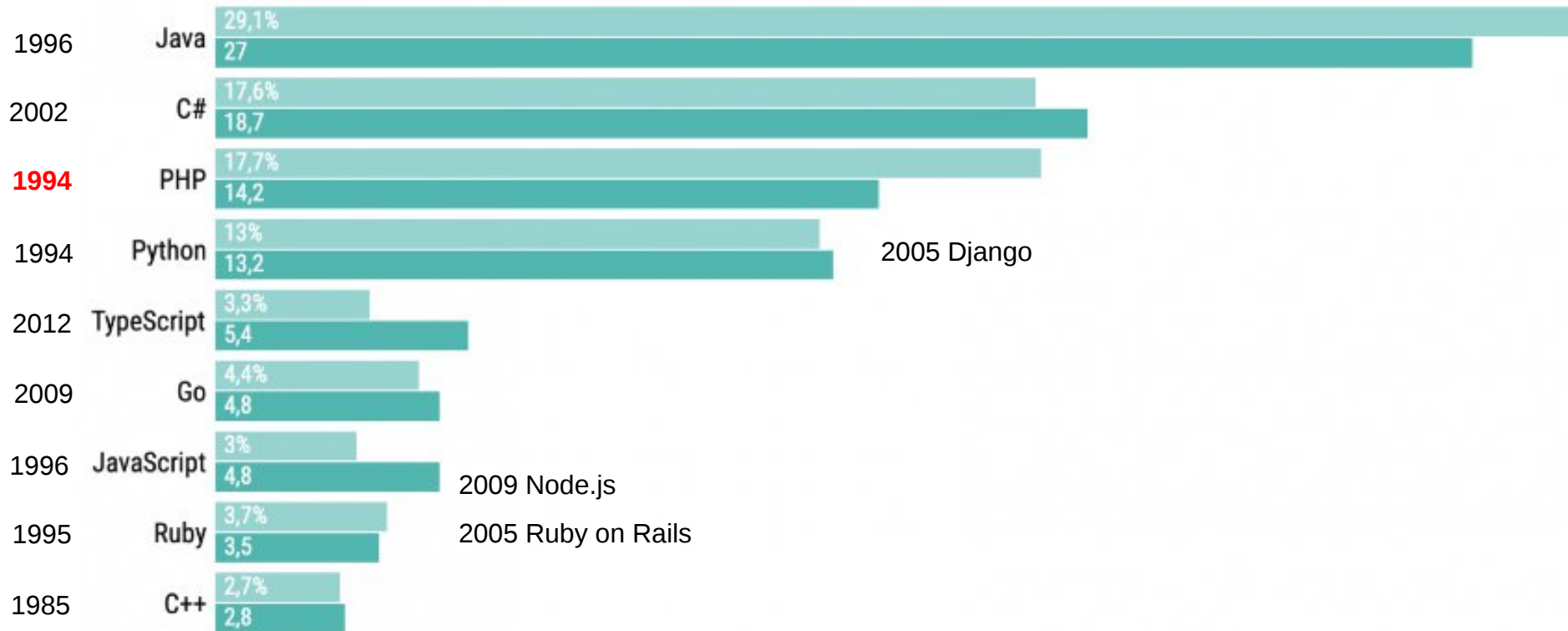
История (необычная) РНР

Backend сегодня



Backend сегодня

Год
создания



Как все начиналось

1994 год от Р.Х.

- Люди общаются и обмениваются файлами в сетях Usenet, FidoNet, Bitnet
- В World Wide Web 2700 веб-сайтов (веб изобретен в 1989 году, в 1992 году было 10 сайтов)
- Веб-сайты становятся динамическими (спецификация CGI)
- Развивается свободное ПО, вышла операционная система GNU/Linux 1.0.0
- Веб-сервер NCSA HTTPd, браузер NCSA Mosaic
- Языки программирования для веб-сайтов: C (1973), C++ (1985), Perl (1987)

Проблема

Существующие языки и инструменты плохо подходили для веб-технологий

```
#!/usr/bin/perl

$first_name = 'Arya';
$last_name = 'Stark';

print "Content-type:text/html\n\n";
print '<html>';
print '<head>';
print '<title>Hello Word - First CGI Program</title>';
print '</head>';
print '<body>';
print '<h2>Hello $first_name $last_name</h2>';
print '</body>';
print '</html>';
```

Проблема

На существующих языках программировать под веб неудобно

- Разметка HTML формируется в виде строк внутри кода (нельзя автоматически проверить структуру)
- Много деталей, связанных с протоколом HTTP
- Язык C слишком низкоуровневый
- Язык Perl плохо подходил для командной работы

«Perl — это единственный язык, программы на котором выглядят одинаково до и после RSA-шифрования»

«PHP — это малое зло, созданное некомпетентными любителями, а Perl — это громадное и хитроумное зло, созданное опытными и извращёнными профессионалами»

Решение

- Разработка и внедрение шаблонизаторов, позволяющих инвертировать процесс: код вставляется в HTML-разметку, а не наоборот.
- Реализация функциональности работы с HTTP-протоколом в ядре или стандартных библиотеках языка.

Как все начиналось



Расмус Лердорф (Rasmus Lerdorf),
1968 г.р. Датский программист,
живет в Канаде

*Ненавижу программирование. Но я
люблю решать проблемы.
Я ужасный кодер, но скорее всего
лучше чем вы:)*

В 1994 году написал Personal Home Page Tool -- набор скриптов на Perl, обрабатывающих шаблоны HTML-документов, для вывода и учета посетителей на его персональной странице с резюме.

Затем переписал скрипты на C и в июне 1995 году выложил их в открытый доступ.

PHP/FI 1 (Personal Home Page Tools/Forms Interpreter)

Концепция

- Основная цель RНР – максимально простое решение повседневных проблем, возникающих в веб-разработке.
- RНР создавался как специальный инструмент, а не универсальный язык программирования. Шаблонизатор, превратившийся со временем в язык программирования.

«RНР как зубная щетка. Вы пользуетесь ей каждый день, она выполняет свою работу, простой инструмент. Что вы хотите знать про зубные щетки?» Рasmus Лердорф

Проще некуда

```
<?PHP

$firstName = 'Arya';
$lastName = 'Stark';

?>

<html>
  <head>
    <title>Hello Word - First CGI Program</title>
  </head>
  <body>
    <h2>Hello <?= $first_name ?> <?= $last_name ?></h2>
  </body>
</html>
```

Явное разделение кода и HTML-блоков.

В HTML можно вставлять данные из PHP-кода. Это уникальная особенность PHP.

Первые шаги

1996 год - **PHP/FI 2.0**

PHP становится похож на C-подобный веб-ориентированный язык программирования, встраиваемый в HTML-синтаксис:

- Переменные, функции, условия, циклы
- Пользовательские функции
- Автоматическая обработка форм
- Поддержка СУБД

1997 год - **PHP 3.0**

Энди Гутманс и Зив Сураски (Израиль) переписывают парсер PHP. Улучшается синтаксис, поддержка ООП. PHP становится модульным. К проекту присоединяются разработчики со всего мира.

2000 год - **PHP 4.0**

Новый движок Zend Engine. Рост производительности, новые языковые конструкции, поддержка HTTP-сессий, более безопасная обработка входной информации. Библиотека классов PEAR.

Бурный рост популярности в начале 2000-х

- Простота, низкий порог входа.
- Бум веб-студий и небольших компаний в начале 2000-х. PHP - единственный инструмент для быстрого и эффективного создания сайтов.
- Все хостинги стали поддерживать стек LAMP из бесплатного ПО с открытым исходным кодом: **L**inux (ОС) + **A**pache (HTTP-сервер) + **M**ySQL (СУБД) + **P**HP (серверный ЯП).
- PHP идеально подходит для хостинга множества мелких сайтов на одном сервере.
- Распространение CMS, написанных на PHP, для создания сайтов без программирования.

Дальнейшее развитие

2004-2014 годы - **PHP 5.0-5.6**

Ядро Zend Engine 2. Развитие ООП. Возможности ФП. Развитие экосистемы: фреймворк для тестирования PHPUnit, пакетный менеджер Composer, стандарты кодирования PSR. Появление множества фреймворков для веб-разработки.

2015 год - **PHP 7.0**

Ядро Zend Engine 3. Новые синтаксические фиши. Контроль типов данных.

2020 год - **PHP 8.0**

Поддержка JIT-компиляции. Новый движок Zend Engine. Рост производительности. Улучшения синтаксиса.

Родовые травмы и особенности

- Непоследовательность синтаксиса языка. Сложно запоминать имена.
- Путаница с передачей параметров. Сложно запоминать сигнатуры функций.
- Странная система предупреждений об ошибках. Fatal, Warning, Notice.
- Возможность прямой загрузки кода без пространств имен.
- Возможность смешать прикладную логику, данные и отображение. Язык не препятствует написанию спагетти-кода.
- "PHP рожден, чтобы умирать"

Снижение популярности

- Появились инструменты (шаблонизаторы, фреймворки) для веб-разработки на других языках популярных языках. Django, Ruby on Rails, ASP.NET
- Проекты становятся большими и сложными. Важнее не порог входа и скорость разработки, а качество и поддерживаемость кода.
- PHP плохо подходит для создания долгоживущих приложений (набора сервисов) и работы с асинхронным кодом.
- Появилось много low code платформ для создания сайтов без программирования. Отказ малого бизнеса от создания собственных сайтов с помощью команды программистов.

Перспективы

- РНР - это большая развивающаяся экосистема, отлично решающая задачи в своей нише.
- Огромное количество текущих проектов на РНР, большое количество вакансий даже для новичков.

PHP как инструмент изучения технологий веб-разработки

- Легковесность, простота развертывания.
- Кроссплатформенность.
- Поддержка процедурного и функционального подхода к программированию.
- Полноценная поддержка ООП на классах.

- Веб внутри языка - максимальная простота вхождения в тему
- Полноценная экосистема
- Современный подход к разработке веб-приложений

Задачи курса

Задачи курса - теория

- Рассмотреть реализацию парадигм программирования в PHP
- Повторить схему и механизмы работы сайтов и веб-приложений
- Сравнить PHP с другими языками и инструментами для веба
- Обсудить эволюцию подходов к разработке веб-приложений на PHP

Задачи курса - практика

- Закрепить навыки работы с Git и GitHub
- Освоить инструменты экосистемы PHP
- Реализовать игровой алгоритм в виде PHP-приложений разных типов:
 1. Консольное без базы данных
 2. Веб-приложение с базой данных на стандартных возможностях PHP
 3. Веб-приложение с базой данных на микрофреймворке Slim
 4. Веб-приложение с базой данных на фреймворке Laravel

Учебные репозитории: <https://github.com/andpop-mrsu>