



История возникновения PHP

В 1998 году появился PHP — язык, специально созданный для работы в Интернете, он получил широчайшее распространение. Сам проект начался ранее, в 1994 году. Первые две версии представляли собой варианты, написанные на Perl датским программистом Расмусом Лердорфом. Язык имел мало общего с современным PHP и больше походил на то, что сейчас принято называть языками шаблонов (вроде Smarty или HAML).



Изначально PHP переводилось как «Personal Home Page Tools» (Инструменты для создания персональных веб-страниц). В настоящий момент PHP переводится как препроцессор гипертекста.



До 5 версии PHP в нем не существовало поддержки концепции объектно-ориентированного программирования, PHP 5 вышло в 2005 году.



Проблемы огромных проектов, написанных на PHP

Низкая производительность

Огромное потребление оперативной памяти



Компилятор и интерпретатор. В чем разница

- 1) Интерпретатор берет одну инструкцию, транслирует и выполняет ее, а затем берет следующую инструкцию. Компилятор же транслирует всю программу сразу, а потом выполняет ее.
- 2) Компилятор генерирует отчет об ошибках после трансляции всего, в то время как интерпретатор прекратит трансляцию после первой найденной ошибки.
- 3) Компилятор по сравнению с интерпретатором требует больше времени для анализа и обработки языка высокого уровня.
- 4) Помимо времени на обработку и анализ, общее время выполнения кода компилятора быстрее в сравнении с интерпретатором.



Мощный, быстрый и другой PHP. PHP7

Успехи социальных сетей в области создания альтернативных реализаций PHP побудило сообщество к выпуску новой, более эффективной версии PHP 7 на новом движке Zend Engine 3, релиз которой состоялся в декабре 2015 года. Титанические усилия команды разработчиков позволили ускорить движок PHP 7 в два раза, на фоне снижения потребления оперативной памяти. В версии PHP 7 значительно улучшена поддержка 64-битных операционных систем.



Переменные в PHP

Каждое имя переменной должно начинаться со знака \$, так нужно для корректной работы интерпретатора.

В официальной документации сказано, что имя переменной может состоять не только из латинских букв и цифр, но также и из любых символов, код которых старше 127, — в частности, и из "русских" букв! Однако мы категорически не советуем вам применять кириллицу в именах переменных — хотя бы из-за того, что в разных кодировках ее буквы имеют различные коды. Стоит отметить то, что имя переменной не может начинаться с цифры.



Типы данных

- Integer (целое число)
- Double (вещественное число)
- String (строка)
- Array (ассоциативный массив)
- Object (ссылка на объект)
- Resource
- Boolean
- Null
- Callable



Целые числа

Целое число со знаком, размер которого зависит от разрядности PHP. В 32-битном варианте целое число может принимать значение от $-2\,147\,483\,648$ до $2\,147\,483\,647$. В 64-битном от $-9\,223\,372\,036\,854\,775\,807$ до $9\,223\,372\,036\,854\,775\,807$. Выяснить, максимальное значение для целого числа в PHP можно, обратившись к предопределенной константе `PHP_INT_MAX`.



Вещественные числа

Числа с плавающей точкой имеют две формы записи. Обычная форма совпадает с принятой в арифметике, например, 346.1256.

Экспоненциальная форма позволяет представить числа в виде произведения мантиссы 3.461256 и соответствующей степени числа 10.

Для цифр меньше нуля степень числа 10 является отрицательной. Так, число 0.00012 в экспоненциальной форме может быть записано как 1.2

Строки

В отличие от C, строки могут содержать в себе так- же и нулевые символы, что никак не повлияет на программу. Иными словами, строки можно использовать для хранения бинарных данных. Длина строки ограничена 2 Гбайт. Однако этот предел редко достигается, т. к. каждый PHP-скрипт ограничен в объеме потребляемой памяти директивой `limit`, который по умолчанию составляет 128 Мбайт.



Ассоциативные массивы

Это набор из нескольких элементов, каждый из которых представляет собой пару вида ключ=>значение (символом => мы обозначаем соответствие определенному ключу какого-то значения). Доступ к отдельным элементам осуществляется указанием их ключа. В отличие от С-массивов, ключами здесь могут служить не только целые числа, но и любые строки.



Ресурс

Некоторый ресурс, который PHP обрабатывает особым образом. Пример ресурса — переменная, содержащая дескриптор открытого файла. Такая переменная может в дальнейшем быть использована для того, чтобы указать PHP, с каким файлом нужно провести ту или иную операцию (например, прочитать строку).



Boolean

Логическая переменная может содержать одно из двух значений: `false` (ложь) или `true` (истина). Вообще, любое ненулевое число (и непустая строка), а также ключевое слово `true` символизирует истину, тогда как `0`, пустая строка и слово `false` — ложь. Таким образом, любое ненулевое выражение (в частности, значение переменной) рассматривается в логическом контексте как истина. Вы можете пользоваться константами `false` и `true` в зависимости от логики программы.



Null

Переменной можно присвоить специальную константу `null`, чтобы пометить ее особым образом. Тип этой константы — особый и называется также `null`. Это именно отдельный тип, и функция `gettype()`, которую мы вскоре рассмотрим, вернет для `null` переменной слово `null`.



Есть же и другие???

Да, действительно сегодня мы не рассмотрим два типа: Object и Callable. Но они обязательно будут рассмотрены на следующих заданиях (когда перейдем к ООП)



Действия с переменными

- Проверка на инициализация переменной
- Уничтожение переменной
- Определение типа переменной
- Приведение переменной к определенному типу данных



Типы ссылок

- Жесткие ссылки
- Символическая ссылка



Что происходит, когда переменная получает новое значение

- Выделяется оперативная память для хранения значения

РНР регистрирует в своих таблицах новую переменную, с которой связывает выделенный участок памяти

