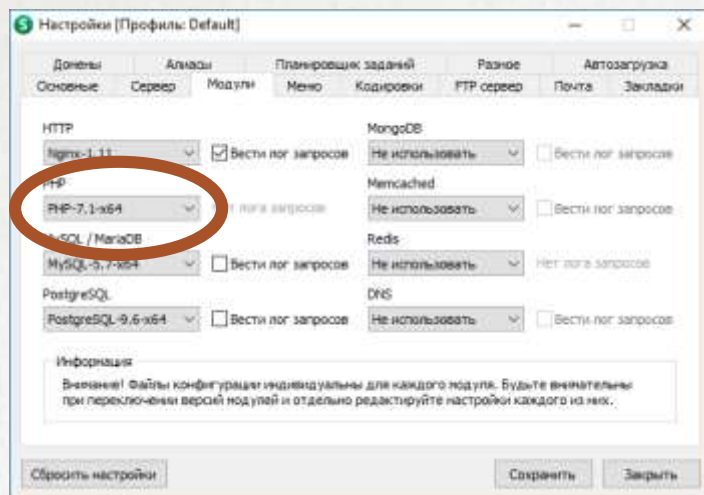


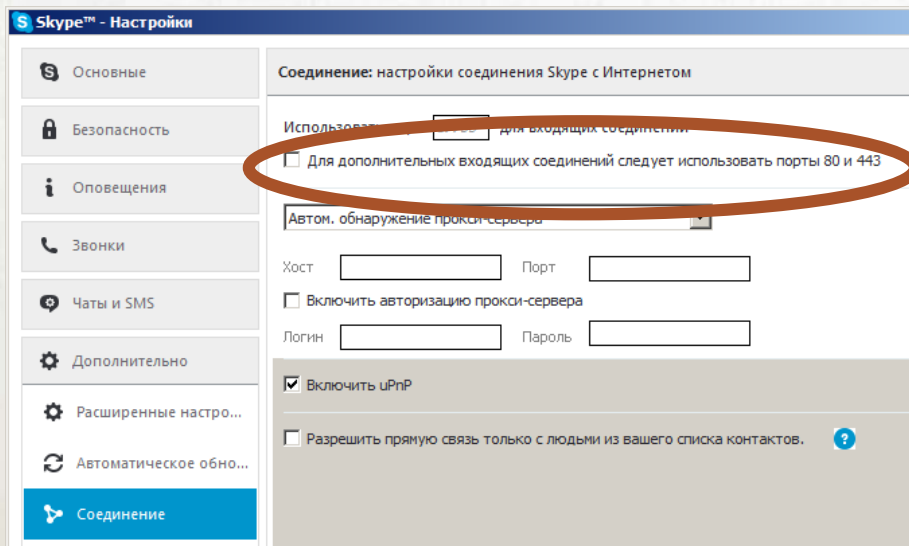
PHP: ВВЕДЕНИЕ В ПРОФЕССИЮ

УРОК 1. БЫСТРЫЙ СТАРТ.

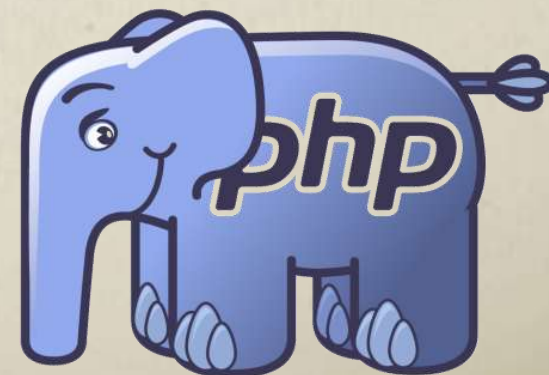
- **PHP** – работает на сервере. Сервером может быть как некий удалённый компьютер, так и программа на вашем компьютере, это неважно.



- **NB.** Если вы используете Skype – заставьте его освободить порт 80!



НАСТРАИВАЕМ СЕРВЕР



PHPStorm – уже, фактически, отраслевой стандарт.

Сложно найти адекватную замену этой среде разработки.

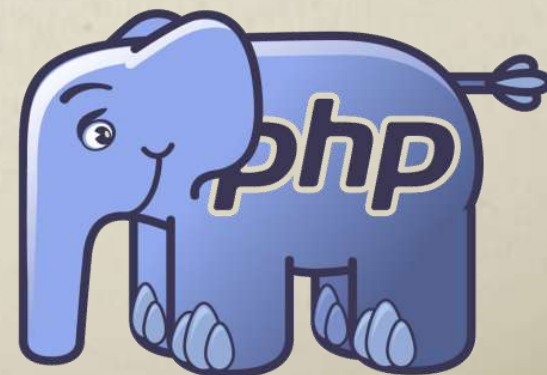
Старайтесь сразу привыкнуть к работе в IDE:

- Говорить с коллегами на одном языке
- Повышать **эффективность** своей работы
- Допускать меньше ошибок

NB. Есть студенческие лицензии, лицензии для разработчиков открытых проектов и другие возможности...

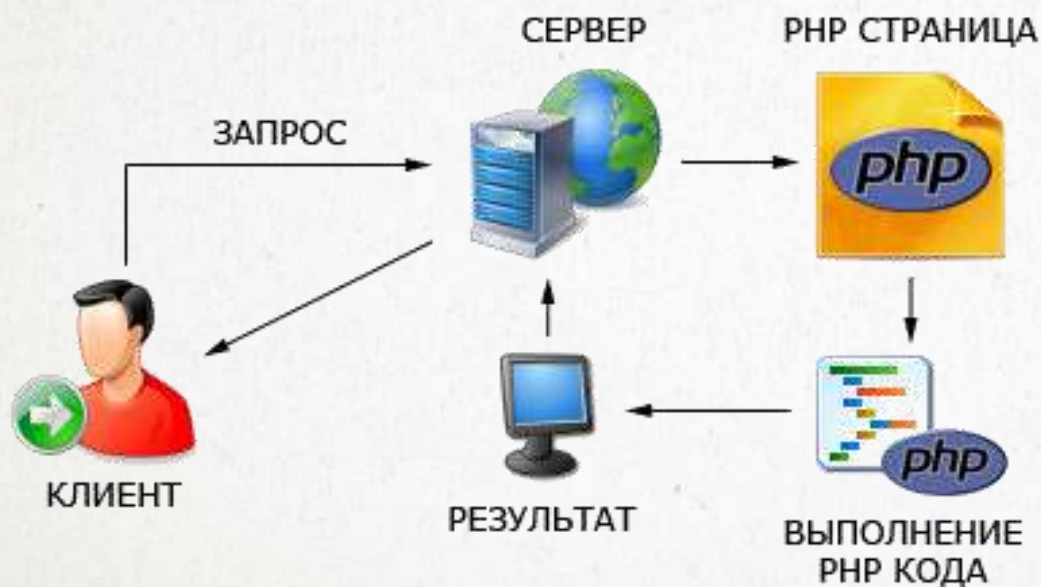


**НАСТРАИВАЕМ
РАБОЧУЮ
СРЕДУ**



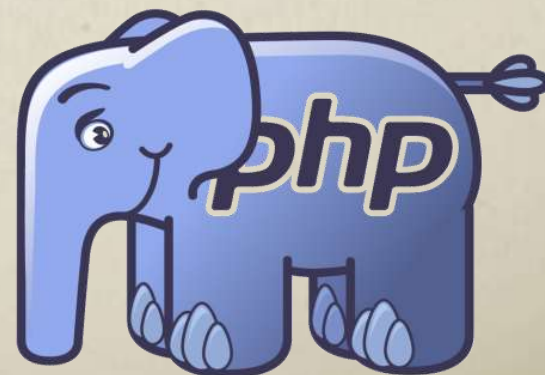
- **PHP** – это язык программирования. Не разметки, как HTML и не стилей, как CSS, а «настоящего» программирования
- Он предназначен для создания бизнес-логики вашего веб-приложения

PHP работает на сервере!



РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ НА PHP – ЭТО ТЕКСТ,
который передается клиенту в ответ на его запрос

ГДЕ И КАК РАБОТАЕТ PHP?



ВЫВОД или как увидеть работу программы?

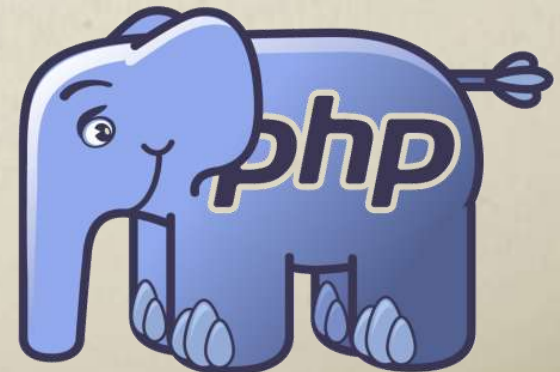
- Любое выражение можно вывести с помощью конструкции **echo**.

При этом вы должны понимать, что вывод будет встроен в страницу, которая, после полной готовности, будет отправлена клиенту:

```
<html>
  <body>
    <p>
      Результат подсчета:
      <?php
        echo 2 * 2;
      ?>
    </p>
  </body>
</html>
```

NB. Не забудьте, что файл должен иметь расширение “.php”, иначе чудя не произойдет!

ПЕРВАЯ ПРОГРАММА ☺



ВЫРАЖЕНИЯ

- **Выражение** – это некое значение (число, строка и т.д.), заданное в явном виде, или в виде вычислений. Например:

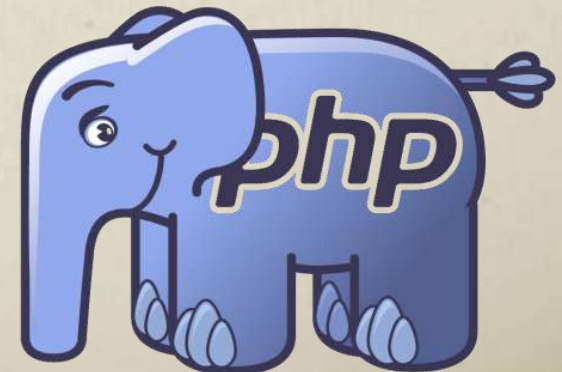
```
2      // это число
1.5    // это тоже число, но нецелое
'foo'  // это строка из 3 символов
2+2    // это выражение, его результат =4
2*2    // не поверите, но тоже 4!
```

- Выражение имеет **значение** (то, чему оно равно)

Операции (операторы)

- Арифметические (+ - * /)
- Логические (&& || !)
- Сравнения (== != < >)
- Строковые
- Битовые
- Работы с массивами, проверки типа и другие, специфичные для PHP

ВЫРАЖЕНИЯ



ПЕРЕМЕННЫЕ

- Переменная – это некое значение (выражение), сохраненное под понятным для нас именем. Например:

```
$email = 'test@example.com';  
$age = 42;  
$result = 2 + 2 * 2;
```

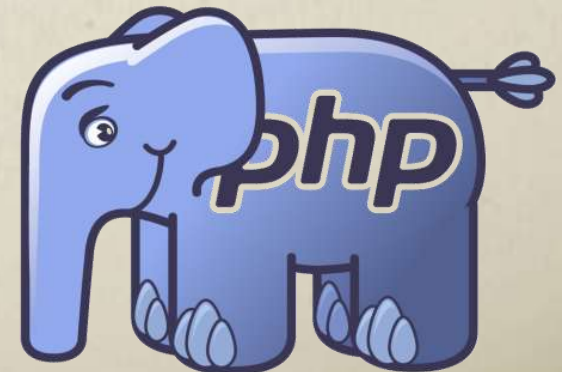
- Имя переменной в PHP начинается со знака «\$». Обратите внимание, что \$foo и \$FOO – это разные переменные!
- В PHP объявление переменной (создание имени) и присваивание ей значения – это одна операция.
- В принципе возможно и просто объявление переменной, без всякого значения:

```
$a;
```

но это слегка бессмысленно 😊

Говорят, что в этом случае переменная имеет значение **null**

ПЕРЕМЕННЫЕ



ТИП – это определение того, может быть значением

В PHP тип имеют значения, а не переменные. Переменная – это просто имя для значения, чтобы его сохранить на будущее.

Самое интересное, что тип выводится автоматически!

- **int, integer**

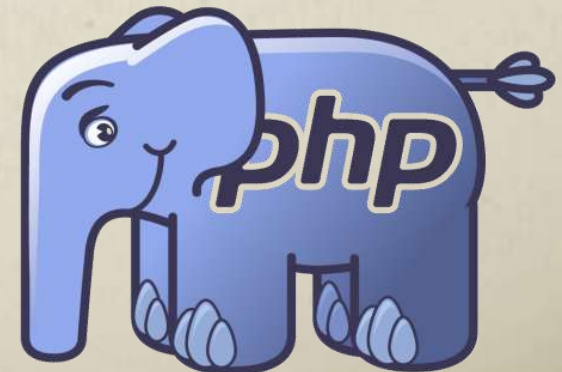
Целые числа. Любые. Включая, конечно, ноль.

Замкнуты относительно сложения, вычитания, умножения – результат снова будет целым числом. А вот при делении – не факт!

- **float**

Тоже числа. Но не целые, а имеющие дробную часть. Иногда нулевую, но всё равно имеющие. Являются приближенными. Легко получить, написав число с десятичной точкой, или используя операции, дающие нецелый результат (деление, как пример)

ТИПЫ



ТИП – это определение того, может быть значением

В PHP тип имеют значения, а не переменные. Переменная – это просто имя для значения, чтобы его сохранить на будущее.

Самое интересное, что тип выводится автоматически!

- **string**

Строки. То, что состоит из символов. В PHP строки:

- Могут быть любой длины (хоть вся «Война и мир»)
- Могут быть нулевой длины (пустые)

```
$str = '';
```

- Могут заключаться в одинарные или двойные кавычки. Это почти одно и то же.

```
$str1 = 'test';
```

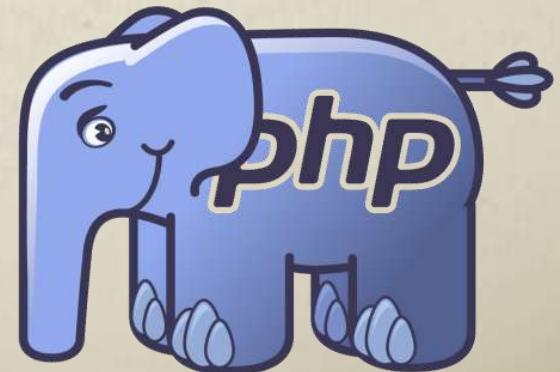
```
$str2 = "test";
```

но:

```
$foo = 'Hello\n';
```

```
$bar = "Hello\n";
```

ТИПЫ



ТИП – это определение того, может быть значением

Тип результата операции всегда определяется оператором!

Например, арифметические операции всегда производят числа:

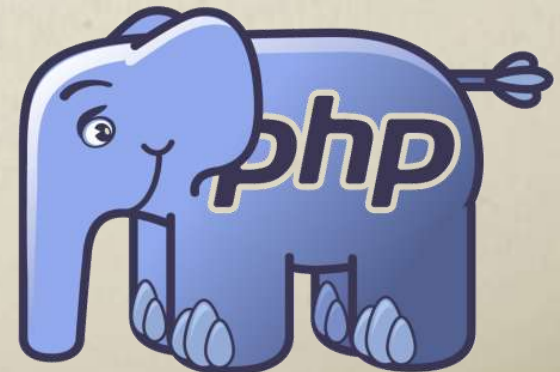
```
$a = $x + $y;    // всегда число!  
$a = 2 + '2';    // 4  
$a = '2' * '2';  // тоже 4
```

А оператор «точка» (сложение строк) всегда производит строки:

```
$a = $x . $y;    // всегда строка!  
$a = 2 . 2;      // '22'  
$a = '2' . 2;    // тоже '22'
```

Этот механизм называется «приведением типов». Тип значения приводится к наиболее подходящему, чтобы соответствовать типу оператора.

ТИПЫ



ФУНКЦИЯ – это подпрограмма, которая принимает на вход параметры (аргументы) и возвращает результат

Одна из самых сильных сторон PHP – богатая стандартная библиотека функций

Функции можно использовать везде, где только можно использовать выражение.

Для использования функции нужно написать ее имя, а затем, в круглых скобках, список аргументов через запятую. Список можно быть и пустым!

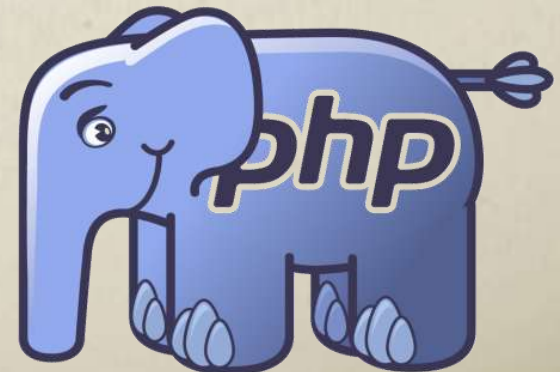
```
echo strlen('Hello');           // 5
$a = 2 + sqrt(2+2);             // 4.0

$x = str_replace('e', 'a', 'Hello');
    // 'Hallo'

'My number is ' . rand();
```

N.B. Невозможно знать наизусть ВСЕ функции стандартной библиотеки. Но возможно научиться пользоваться мануалом!

ФУНКЦИИ



ДО ВСТРЕЧИ НА СЛЕДУЮЩЕМ УРОКЕ!

**ВИДЕОЗАПИСЬ, СЛАЙДЫ, ПРЕЗЕНТАЦИЯ
И ДОМАШНЕЕ ЗАДАНИЕ
БУДУТ ВЫЛОЖЕНЫ ДО 10 УТРА СЛЕДУЮЩЕГО ДНЯ**

