

Институт повышения квалификации и переподготовки кадров Учреждения образования «Гродненский государственный университет имени Янки Купалы»

**Технологии компонентного программирования**



# ОСНОВЫ РНР



## Технологии компонентного программирования

- ✓ PHP – это гибкий и легкий язык web-программирования
- ✓ PHP это язык программирования, с помощью которого создают сайты, активно взаимодействующие с пользователем
- ✓ PHP отличается от других подобных языков (JavaScript), тем, что код выполняется на сервере.
- ✓ Наилучшим качеством PHP является то, что он прост для новичка в программировании и предлагает много продвинутых возможностей



## Технологии компонентного программирования

✓ 1994 год –  
программист Расмус  
Лерддорф написал  
собственные Perl-  
скрипты для  
собственной  
страницы и назвал  
Personal Home Page  
(PHP)







## Технологии компонентного программирования



**1997 год** - два программиста Энди Гутманс и Зив Сураски взяли за основу идею Расмуса и переписали PHP с нуля. Название "Personal Home Page" было изменено на **Hypertext Preprocessor**





### Основные способы вставки PHP кода

PHP код состоит из инструкций, разделенных знаком ;

Правильные записи

```
<?php  
    инструкция1;  
    инструкция2;  
?>
```



### Переменные в PHP

- ✓ PHP создан не только для форматирования статичного текста. Для того, чтобы обрабатывать различные данные были придуманы **переменные**.

Например:

- `<?php $name = "Виктор"; ?>`





## Технологии компонентного программирования

### Переменные в PHP

- ✓ начинаются со знака доллара (\$);
- ✓ могут содержать только латинские буквы, цифры, символы подчеркивания;
- ✓ состоят из одного или нескольких символов и начинаются с буквы или символа подчеркивания;
- ✓ не может совпадать с каким-либо из стандартных ключевых слов PHP;
- ✓ учитывается регистр символов.



# Вывод текста на экран.

## Оператор echo.

Когда нужно отобразить текст на веб-странице, то оператор echo является наиболее употребляемым оператором в PHP. Как его использовать - после слова echo нужно поместить строку текста в кавычки:

```
<?php  
echo 'Привет от PHP';  
?>
```





### Функции ввода

В PHP существует несколько способов вывода информации на страницу:

Оператор `print` выводит значение аргумента:

```
print (<строка>);
```

```
var_dump
```

```
(<переменная>); - выводит переменную вместе с типом.
```



## Технологии компонентного программирования

Отображение в браузере:

Привет от PHP

(для отображения текста можно использовать как двойные кавычки, так и одинарные).

Для чисел кавычки можно не использовать:

```
<?php
```

```
echo 2014;
```

```
?>
```



## Технологии компонентного программирования

Знак	Назначение	Пример
+	Сложение двух переменных.	$\$v = 1; \$k = 2; \$s = \$v + \$k; // \$s = 3$
-	Вычитание одной переменной из другой	$\$v = 1; \$k = 2; \$s = \$v - \$k; // \$s = -1$
*	Произведение переменных	$\$v = 1; \$k = 2; \$s = \$v * \$k; // \$s = 2$
/	Деление одной переменной на другую	$\$v = 1; \$k = 2; \$s = \$v / \$k; // \$s = 0.5$
%	Остаток от целочисленного деления	$\$v = 1; \$k = 2; \$s = \$v \% \$k; // \$s = 1$
++	Инкремент (увеличение на единицу)	$\$v = 6; \$s = \$v++; // \$s = 6, \text{ а } \$v = 7$ $\$v = 6; \$s = ++\$v; // \$s = 7, \text{ и } \$v = 7$





## Технологии компонентного программирования

Знак	Назначение	Пример
--	Декремент (уменьшение на единицу)	$\$v = 6; \$s = \$v--; // \$s = 6, a \$v = 5$ $\$v = 6; \$s = --\$v; // \$s = 5, и \$v = 5$
• точка	Сложение двух строк (конкатенация)	$\$v = "Как"; \$k = "Вас"; \$m = "зовут?";$ $\$s = \$v . " " . \$k . " " . \$m; // \$s = "Как Вас зовут?"$
? =	Составное присваивание. Вместо знака ? должен быть подставлен один из знаков операций: + - * / % .	$\$a = 3; \$b = 1; \$b += \$a; // \$b = 3$ $\$a = 2; \$b = 1; \$b -= \$a; // \$b = -1$ $\$a = 3; \$b = 2; \$b *= \$a; // \$b = 6$ $\$a = 5; \$b = 1; \$b /= \$a; // \$b = 0.2$ $\$a = "Здравия"; \$b = "Желаю";$ $\$b .= \$a; // \$b = "Желаю Здравия"$ $\$a = 5; \$b = 17; \$b \% = \$a; // \$b = 2$



## Технологии компонентного программирования

### Логические операторы

Обозначение	Название	Пример
<b>and &amp;&amp;</b>	И	\$a and \$b \$a && \$b
<b>or   </b>	ИЛИ	\$a or \$b \$a    \$b
<b>xor</b>	Исключающее ИЛИ	\$a xor \$b
<b>!</b>	Инверсия (NOT)	! \$a



## Технологии компонентного программирования

### Однострочные

```
<?php
    // Это комментарий
    # Это тоже комментарий
?>
```

```
<?php
    /* Это комментарий
    многострочный
    echo "Привет";
    */
?>
```





### Создание переменной

Переменная создается тогда, когда ей присваивают какое-нибудь значение. Для присвоения значения переменной используют оператор присвоения.

Например:

```
<?php
```

```
$surname = "Петров";
```

```
$number = 1269794645;
```

```
$pi = 3.14159265;
```

```
$hello = "Hi all";
```

```
?>
```



**Переменную можно вывести на экран с помощью оператора echo, вот так:**

```
<?php  
$name = "Виктор";  
echo "Ваше имя ", $name, "<br>";  
?>
```

Отображение в браузере:

Ваше имя Виктор



## Технологии компонентного программирования

Создадим переменную которая будет содержать значение количества бананов, вторая переменная количество лимонов, а третья - их суммарное количество.

```
<?php
$bann = 5; // Бананы
$lim = 10; // Лимоны
$together = $bann + $lim; // Всего
echo "Количество фруктов ", $together;
?>
```

Отображение в браузере:

Количество фруктов 15





# Типы данных PHP

PHP является языком динамической типизации (тип переменной определяется на основе её значения).

**Типы, которые можно использовать в PHP:**

- ✓ **Boolean.** Это логический тип, который содержит значение TRUE или FALSE.
- ✓ **Integer.** Содержит значения целого числа (Например: 4 или 10 или другое целое число).
- ✓ **String.** Содержит значение текста произвольной длины (Например: Олег, Киев, Австрия).
- ✓ **Float.** Вещественное число (Например: 1.2, 3.14, 8.5498777).
- ✓ **Object.** Объект.
- ✓ **Array.** Массив.
- ✓ **Resource.** Ресурс (Например: файл).
- ✓ **NULL.** Значение NULL.



## Технологии компонентного программирования

### Функции для работы с переменными:

- ✓ **is\_integer (переменная)** - возвращает true, если *переменная* - целое число;
- ✓ **is\_double (переменная)** - возвращает true, если *переменная* - действительное число;
- ✓ **is\_string (переменная)** - возвращает true, если *переменная* - строка;
- ✓ **is\_numeric (переменная)** - возвращает true, если *переменная* является числом или строковым представлением числа (строка состоящая из цифр);



## Технологии компонентного программирования

### Функции для работы с переменными:

- **is\_bool** (*переменная*) - возвращает true, если *переменная* является логической;
- **is\_array** (*переменная*) - возвращает true, если *переменная* - массив;
- **gettype** (*переменная*) - возвращает строку, соответствующую типу переменной (integer, double, string...).





## Технологии компонентного программирования

Числовые данные обрабатываются при помощи таких операторов :

**+**, **-**, **\***, **/**, **%** (остаток от деления)

```
<?php
```

```
echo "2 + 2 = ", 2 + 2, "<br>";
```

```
echo "5 - 2 = ", 5 - 2, "<br>";
```

```
echo "10 * 10 = ", 10 * 10, "<br>";
```

```
echo "100 / 2 = ", 100 / 2, "<br>";
```

```
echo "10 % 2 = ", 10 % 2, "<br>";
```

```
?>
```

Отображение в браузере

**2 + 2 = 4**

**5 - 2 = 3**

**10 \* 10 = 100**

**100 / 2 = 50**

**10 % 2 = 0**



## Технологии компонентного программирования

Функция `rand ([int start, int end])` возвращает случайное число из диапазона, заданного начальным значением *start* и конечным значением *end*. Если значения не заданы, то возвращается случайное число от 0 до максимально-возможного значения

✓ `$a = rand();`

✓ `$b = rand(1,50);`