

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

Технологии компонентного программирования



ОСНОВЫ РНР



Типы данных

- ✓ Четыре скалярных *типа*;
- ✓ Два смешанных *типа*;
- ✓ Два специальных *типа*.



Тип `boolean` (булев или логический тип)

Ключевые слова **TRUE** (истина) или **FALSE** (ложь) определяют данный тип данных.

```
<?php
    if ($a == False) {echo "Ложь!"; }
    if ($ a == "Ложь!") { echo "Истина"; }
?>
```



Тип integer (целые)

Этот тип задает число из множества целых чисел $Z = \{\dots, -2, -1, 0, 1, 2, \dots\}$

`<?php`

`$a = 1234;` - положительное

`$a = -123;` - отрицательное

`$a = 0123;` - восьмеричное число

`$a = 0x1A;` - шестнадцатеричное число

`?>`



Тип float (числа с плавающей точкой)

Числа с плавающей точкой (они же числа действительные числа)

```
<?php
```

```
    $a = 1.234;
```

```
?>
```



Технологии компонентного программирования

Тип array (массив)

Это набор символов.

Строка может быть определена:

- ✓ с помощью *одинарных кавычек* ;
- ✓ с помощью *двойных кавычек* ;

```
<?php
```

```
echo 'Строка 1';
```

```
echo 'Строка 1';
```

```
?>
```



Смешанные типы

1. Тип `array` (массив) - в PHP представляет собой упорядоченную карту (ключ - значение).

Конструкция:

`array (key => value, key1 => value1, ...)`

`array (value1, value2, ...)`



Технологии компонентного программирования

Пример 1:

```
<?php  
    $lan = array (  
        "1" => "PHP",  
        "2" => "JAVA"  
    );  
    echo $lan["1"];  
    echo $lan["2"];  
?>
```




Технологии компонентного программирования

Пример 2:

```
$arr = array (  
    "a",  
    "b",  
    "c"  
);  
print_r($arr);
```

Результат: `Array ([0] => a [1] => b [2] => c)`



Условные операторы

Оператор if

позволяет выполнять
фрагменты кода в зависимости
от условия

if (выражение) блок выполнения

Если 'блок выполнения' содержит
несколько команд, то он должен быть



Технологии компонентного программирования

Пример

```
$a = 100;  
if ($a == 100) { echo "True"; }
```

```
$a = 100;  
if ($a == 100) {  
    echo "True";  
    $b = 100;  
}
```



Оператор else

Оператор **else** расширяет **if** на случай, если проверяемое в **if** выражение является неверным, и позволяет выполнить какие-либо действия при таких условиях.

if (выражение) блок выполнения 1
else блок выполнения 2



Технологии компонентного программирования

Пример

```
$a = 100;  
if ($a == 100) {  
    echo "True";  
    $b = 100;  
}  
else {  
    echo "False";  
    $b = 200;  
}
```



Оператор `elseif`

Еще один способ расширения *условного оператора* `if`

`if` (выражение) блок выполнения 1

`elseif`(выражение 1) блок выполнения 2

...

`else` блок выполнения N