Amirkabir University of Technology
(Tehran Polytechnic)

Database Systems Fundamentals

Using PHP Language

Arman Malekzade

Amirkabir University of Technology
(Tehran Polytechnic)

Notice: The class is held under the supervision of Dr.Shiri

github.com/arman-malekzade

arman.malekzade@aut.ac.ir

Chapter 3- HTML & PHP

Roadmap :

- Hypertext Markup Language
- Hypertext Preprocessor
  - Variables, Constants, Data Types
  - Arrays
  - Functions
  - Conditional Statements
  - Loops
  - Object-Oriented (Classes)
  - PDO

Hypertext Preprocessor

- Designed primarily for web development
- Originally created by Rasmus Lerdorf in 1994
- The PHP interpreter, is itself written in C
- An HTML-embedded Web scripting language
- Originally stood for "Personal Home Page”
- Zeev Suraski and Andi Gutmans, rebuilt PHP's core, releasing the updated result as PHP/FI 2 in 1997.
- The acronym was formally changed to PHP: HyperText Preprocessor, at this time.

Variables in PHP

- Formal Definition: $var_name=value;
- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and …)
- Variables are case sensitive.

Constants in PHP

- Formal Definition: define(“cons_name”,value);
- A constant is an identifier (name) for a simple value.
- That value cannot change during the execution of the script
- By convention, constant identifiers are always uppercase.
- The name of a constant follows the same rules as any label in PHP.

Constants in PHP: Magical Constants

__FUNCTION__	The function name
__DIR__	The directory of the file
__CLASS__	The class name
__METHOD__	The class method name
__FILE__	The full path and filename of the file with symlinks resolved

PHP Data Types

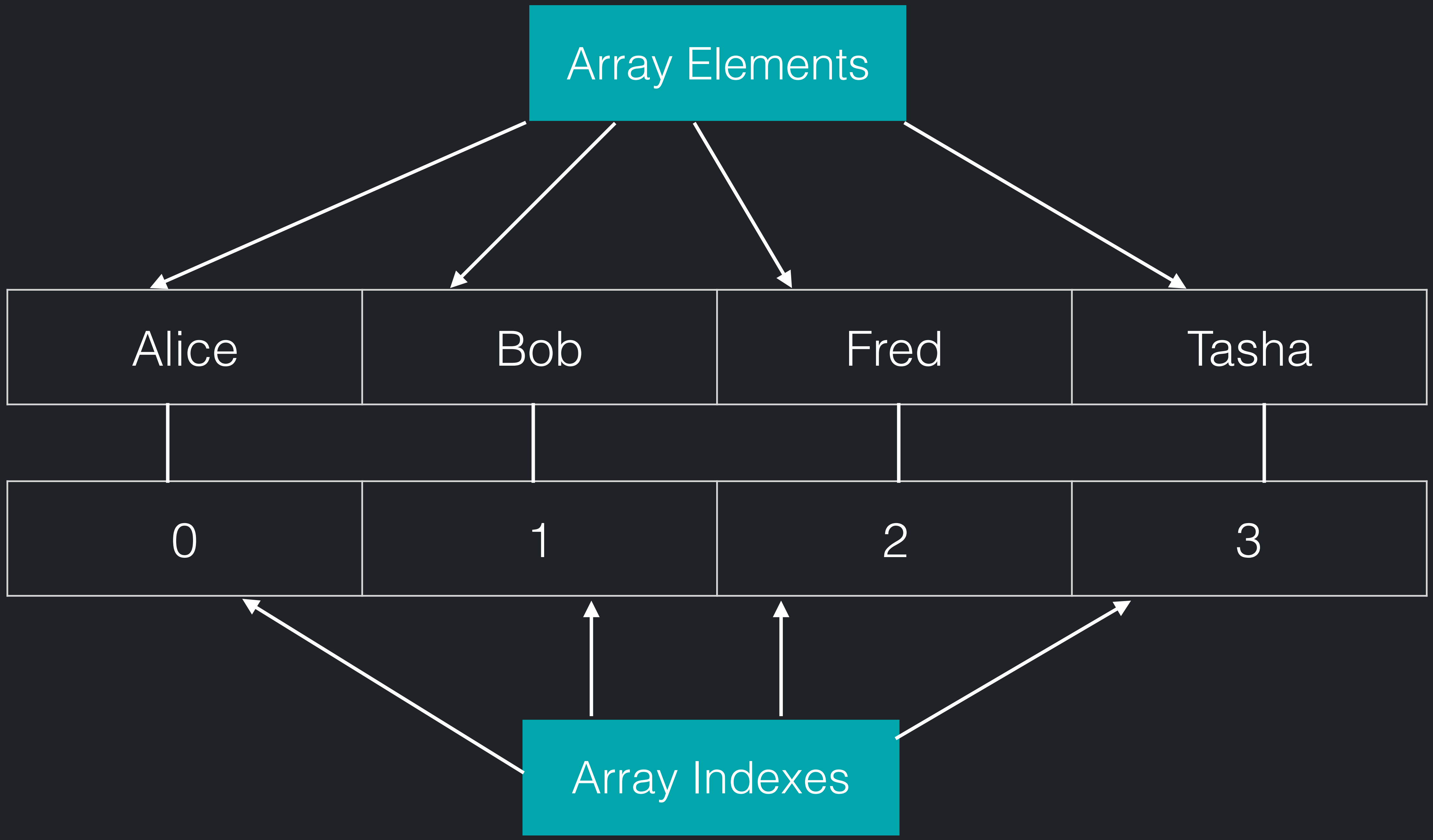
- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

- var_dump is used to find the type of a variable.
  Ex: Code : $txt=‘hello’;var_dump($txt);
  Output: string(5)

Arrays in PHP

- Formal Definition: $arr_name=array(values);
- Ex: $simple_array=array(1,2,3,4,5,6);
- Is equivalent to $simple_array=array(‘0’=>1,’1’=>2,…,’5’=>6);
- Associative arrays can be treated like a hashmap.
- They can also be considered as a stack!
- Ex: array_push($simple_array,”new_value”,”another new value”);
- Ex: array_pop($simple_array);

Arrays in PHP: Associative Arrays



Array Elements

Alice	Bob	Fred	Tasha
0	1	2	3

Array Indexes

Arrays in PHP

- There are 3 kinds of arrays in PHP

1. Numeric (Like C Language)

   Ex: $arr=array(1,2,3,4,5,6);

2. Associative (Hash Tables)

   Ex: $assoc_arr=array(‘name’=>’Tasha’,Code=>15);

3. Multi-dimensional (Like C Language)

   Ex: $md_arr=array(array(1,2,3), array(4,5,6), array(7,8,9));

PHP Functions

- Ex:

Definition:

```
function concat($param1,$param2){ // pass by value
        return $param1.$param2;
}
```

Call : `echo concat(‘hello ’,’world’);`

Definition:

```
function increment(&$param){  // pass by reference
        $param += 1;
}
```

Call: `$x=1; increment(&$x); //x becomes 2`

PHP Conditional Statements

- Example:

```
If($x > 1){
    echo ‘its greater than 1’;
} elseif($x==1){
    echo ‘its equal to 1’;
} else{
    echo ‘its less than 1’;
}
```

```
switch($x){
        case 1:
                echo ‘its 1’;
                break;
        case 2:
                echo ‘its 2’;
                break;
        default:
                echo ‘none’;
}
```

PHP Loops

- Example:

```
while($x>1){
    echo ‘its greater than 1’;
}

do{
    $x=$x+1;
}while($x>1);

for($i=1;$i<10;$i++){
     echo $x;
}
```

```
$arr=array(1,2,3);
foreach($arr as $element){
    echo $element;
}
foreach($arr as $key=>$val){
    echo $key.’   ‘.’$val;
}
```

PHP OP: Classes

- Example:

```
class user{
    protected $name;
    protected $age;
    function user($name,$age){
        $this->name=$name;
        $this->age=$age;
    }
    function set_name($str){
        $this->name=$str;
    }
}
```

```
class student extends user{
    protected $avg;
    function get_avg(){
        return $this->avg;
    }
    function update_avg($avg){
        $this->avg=$avg;
    }
}
```

PHP Data Objects

1. Establish a connection to the server

```
$conn = new PDO(“mysql:host=$servername;dbname=$dbname;charset=utf8", $username, $password);
```

2. Prepare a statement

```
$stmt = $conn->prepare(“INSERT INTO users (name,picture) VALUES (:name,:picture)”);
$stmt = $conn->prepare(“SELECT * FROM $table_name WHERE ID=:id”);
```

3. Bind Parameters

```
$stmt->bindParam(‘:name’,$name);$stmt->bindParam(‘:picture’,$picture);
$stmt->bindParam(‘:id’,$id);
```

4. Execute the Query

```
$stmt->execute();
```

5. Fetch Result(s) [If Needed]

```
$result=$stmt->fetch(PDO::FETCH_ASSOC); //the result is an associative array
```

References

[1] PHP Cookbook - O'Reilly Media
shop.oreilly.com/product/0636920029335.do

[2] Constants
- Retrieved Mar 6, 2017 from
http://php.net/manual/en/language.constants.php

[3] PHP 5 Variables - Retrieved Mar 6, 2017 from
www.w3schools.com/php/php_variables.asp

[4] The History of PHP - Retrieved Mar 6, 2017 from
http://www.nusphere.com/php/php_history.htm