Arrays

HFCC – CIS222

8 February 2012

Announcements:

- ✔ **Quiz 4**
  - You just took it…
- ✔ **Homework 3 (Loops) due on Wednesday**
  - 15 Feb
  - There is a quiz on Loops on that day too…

Why Arrays?

- Variables
  - Hold one thing at a time
  - Putting a value into a variable removes the old value

```
$i = 0;  // $i contains a 0
$i = 2;  // $i now contains a 2, the zero is all gone!!
```

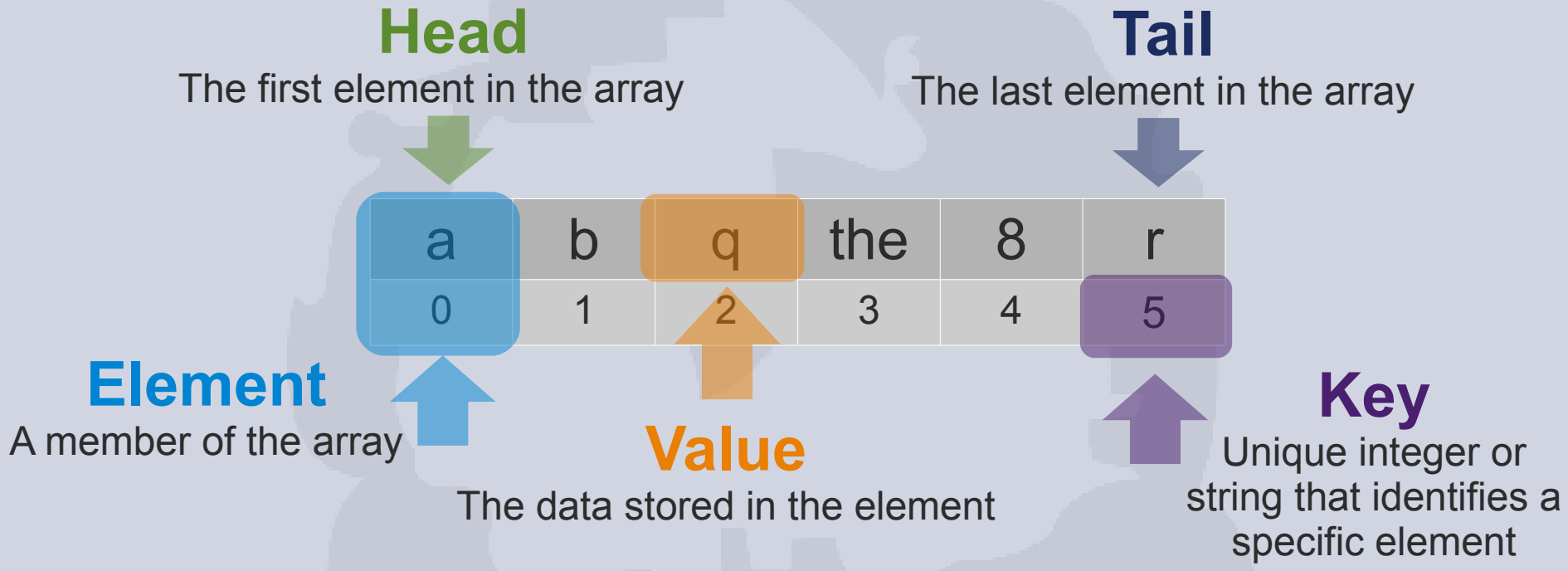
- Arrays
  - Allows a variable to hold multiple values

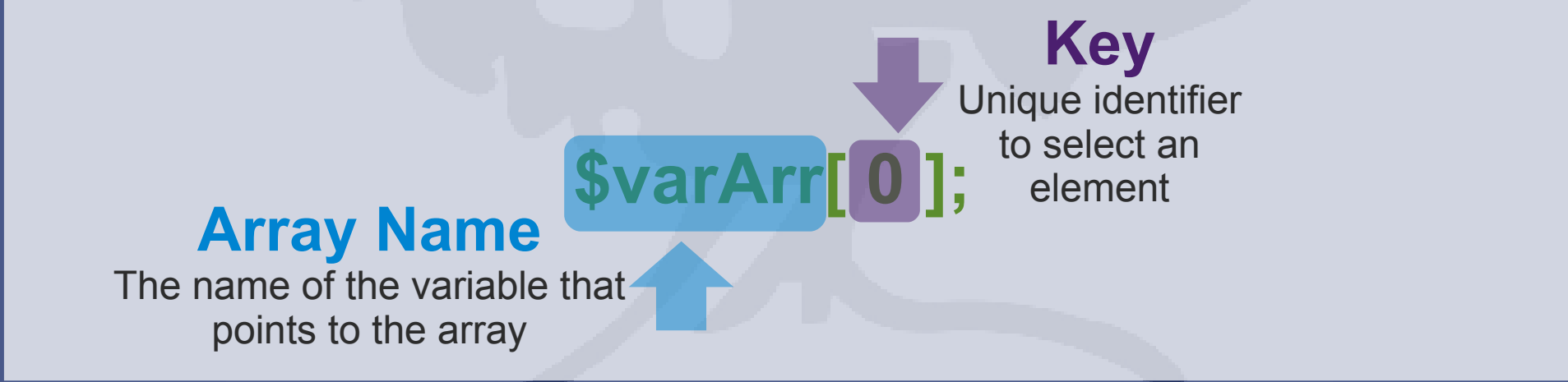
```
$i = array('a', 'b', 'q', 'the', '8', 'r');
```

$i[ ] now contains:

a	b	q	the	8	r

Anatomy of an Array



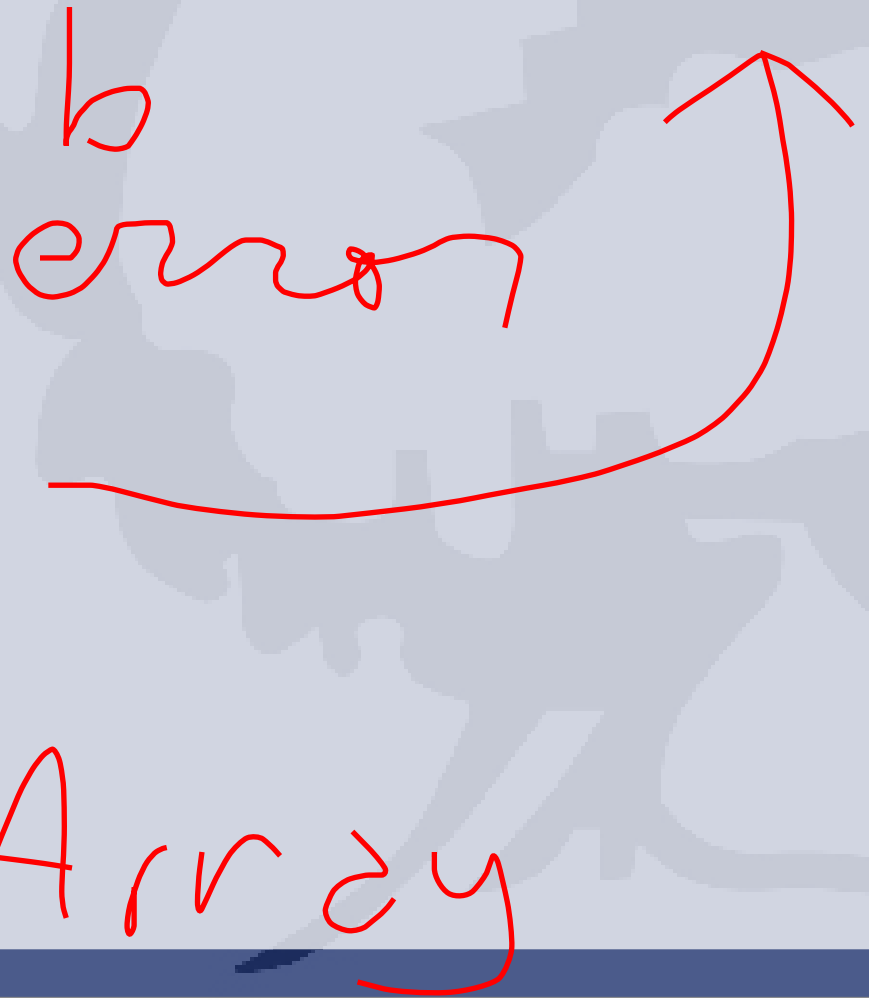


Interacting with Arrays

$arr =

a	b	q	the	8	r
0	1	2	3	4	5

```
echo $arr[1];

echo $arr[6];

echo $arr[3];


$arr[2] = 'blah';


echo $arr;
.
```



Interacting with Arrays

- To see the contents of the array
  - Must use the variable name and the unique key
    - Unique key must be inside of square braces [ ]

```
$arr[2] = 0;   // Places a 0 in the third 'slot' in the array
```

  - Cannot use just the array name
    - It is no longer a variable, it now points to an array

```
echo $arr;   // Will just output the word 'Array'
```

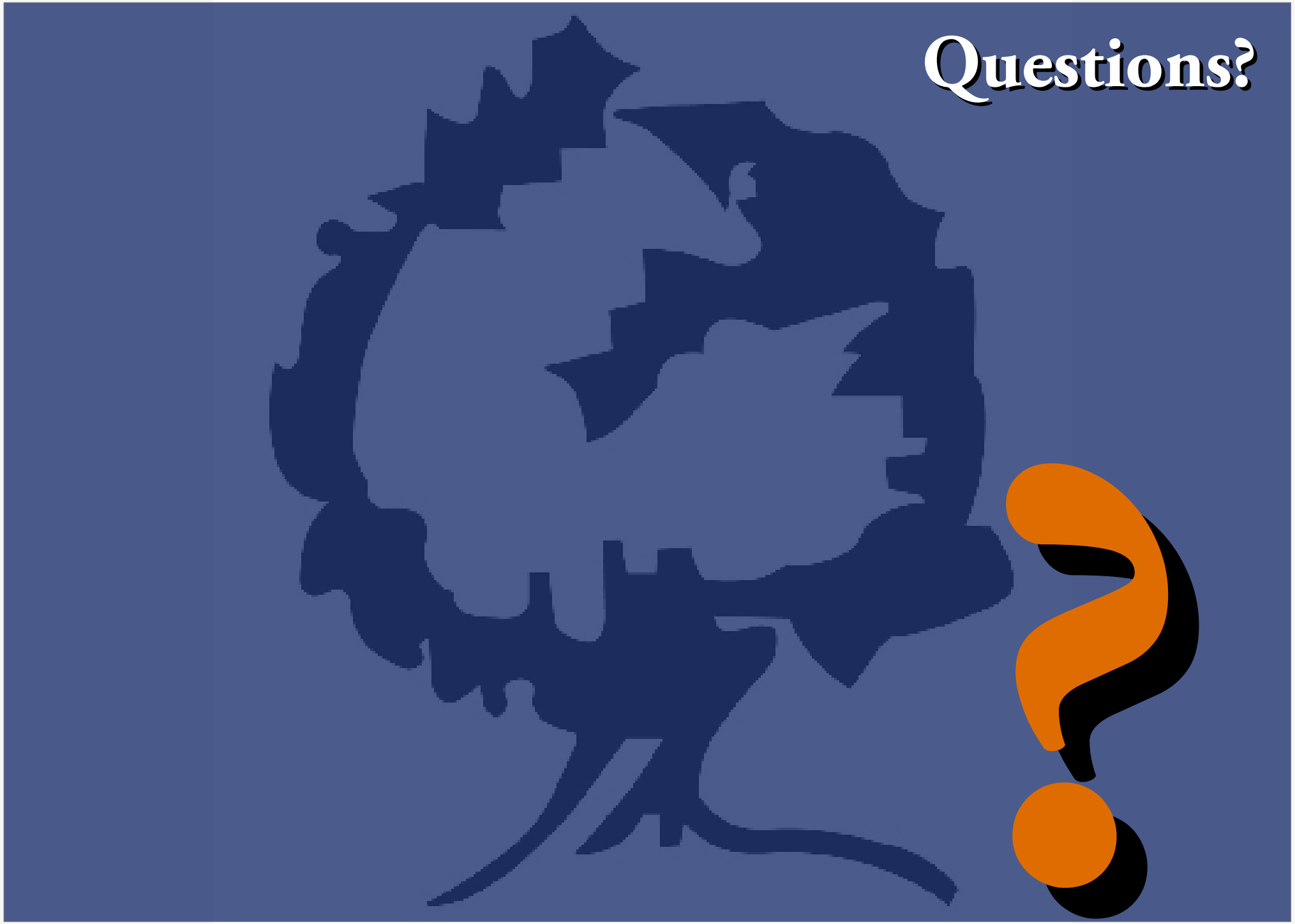
- Interacting with the whole array
  - You can copy arrays as if they are variables

```
$newArr = $arr;
echo $newArr[2];    // Outputs the 0 that we put in above
```

  - You can load an array at declaration
    - Use the 'array( )' function

```
$newerArr = array('r', 'e', 'd', 4);
```

Questions?



Let's Try It!


```
echo $arr[1], $arr[3], $arr[4];

$arr[6] = 'the';

$arr[2] = 'blah';

echo $arr;
.
```

first	second	third	fourth	fifth	sixth
0	1	2	3	4	5

Index vs Position

- Index
  - Refers to the integer key of an element
    - The index of 'second' is 1
    - The index of 'fifth' is 4
    - The first index is always 0
    - The last index is always one less than the count of the elements in the array
- Position
  - The ordinal place where you find an element relative to the head of the array
    - The position of 'first' is 1
    - The position of 'third' is 3
    - The first position is always 1
    - The last position is always the count of the elements of the array

Think in terms of the index being their 'id number' and the position being a 'counting' identifier.

first	second	third	fourth	fifth	sixth
0	1	2	3	4	5

Array Functions

- count($arrayName);
  - Returns the number of elements in the array

```
echo count($arr);   // Outputs: 6
```

- *$value* = array_shift($arrayName);
  - Removes the head of the array (first element)
  - Returns the value of the head

```
$val = array_shift($arr); // $val == 'first'
```

second	third	fourth	fifth	sixth
0	1	2	3	4

- *$value* = array_pop($arrayName);
  - Removes the tail of the array (last element)
  - Returns the value of the tail

```
$val = array_pop($arr);  // $val == 'sixth'
```

second	third	fourth	fifth
0	1	2	3

second	third	fourth	fifth
0	1	2	3

More Array Functions

- array_unshift($arrayName, $value);
  - Inserts a new element as the head of the array

```
array_unshift($arr, '1st');
```

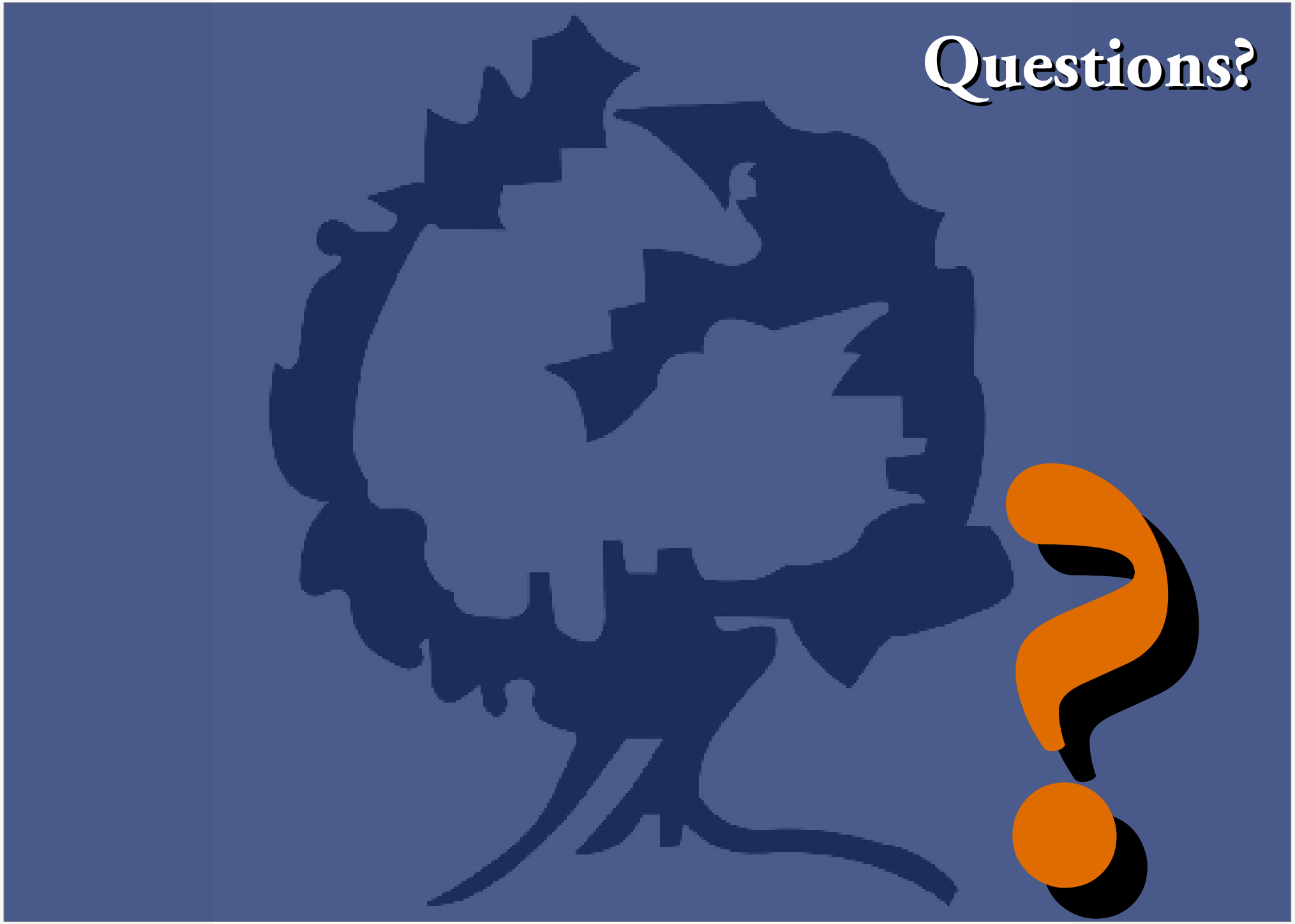
1st	second	third	fourth	fifth
0	1	2	3	4

- array_push($arrayName, $value);
  - Appends a new element to the tail of the array

```
array_push($arr, '6th');
```

1st	second	third	fourth	fifth	6th
0	1	2	3	4	5

Questions?



Let's Try It!

- **Fun and games with arrays!!**
- Create a basic page ***InClass-Arrays.php***
  - Set up an array ***$grades*** with values 75, 80, 95, 100, and 100
    - Output the contents of the array
  - Move the last grade to the beginning of the array
    - Output the contents of the array again
- **Raise your hand if you have finished (or if you are stuck)**

Loops + Arrays = <3

first	second	third	fourth	fifth	sixth
0	1	2	3	4	5

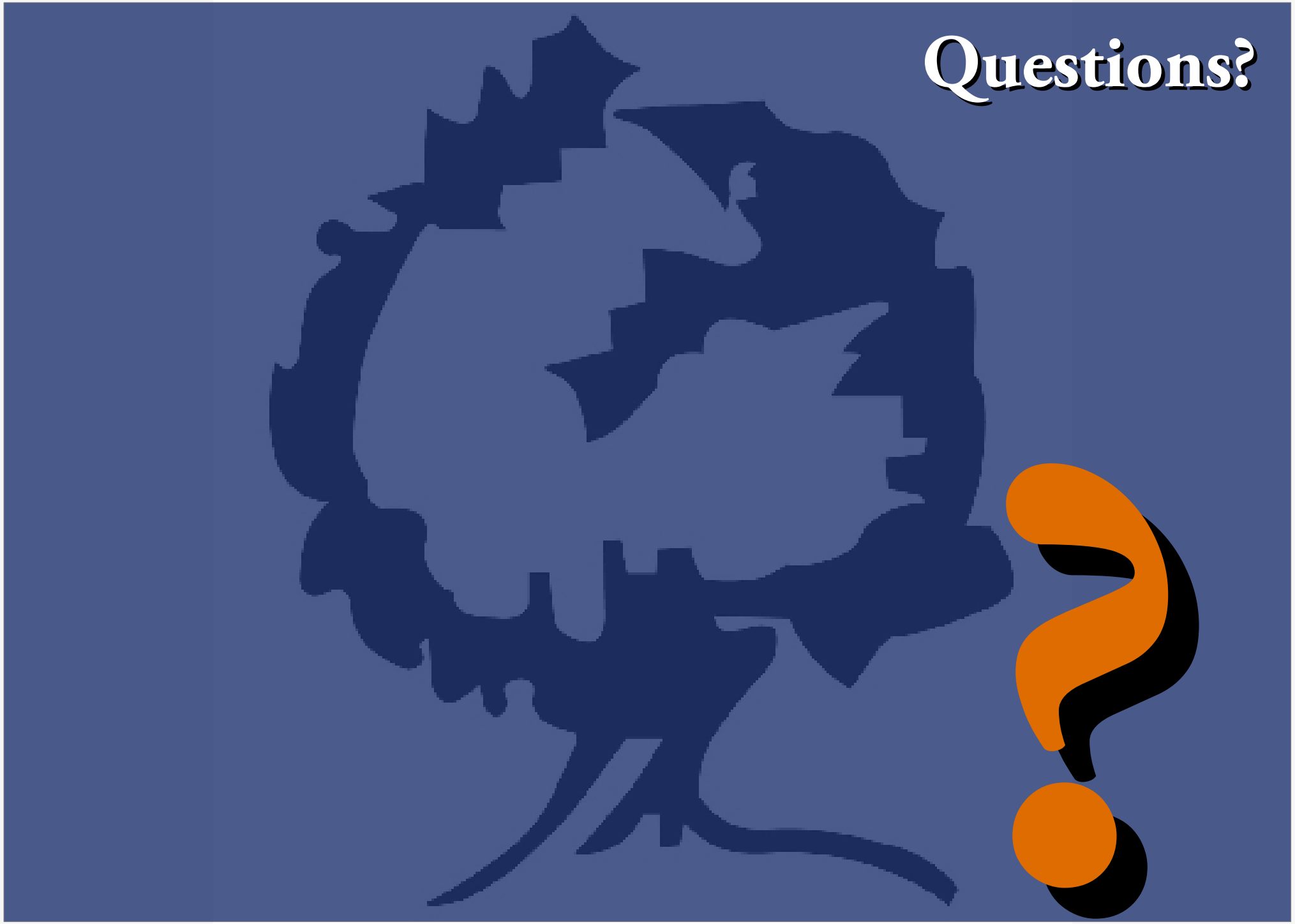
- Arrays are a whole lot more useful with loops

```
for($i = 0; $i < count($arr); ++$i) {
    echo $arr[$i];
    echo '<br>';
}
```

Notes:

- Index of an array always starts at 0 ***($i = 0)***
- You want to loop count($arr) times ***($i < count($arr))***
- Each iteration, $i is an increasing number, so we can use it as the index selector for our output ***($arr[$i])***

Questions?



Let's Try It!

- **More array madness!!!1!!1!one!!!1!**
- Create a basic page ***InClass-ArrayFor.php***
  - Use the array ***$grades*** with values 75, 80, 95, 100, and 100 from last time
  - Output the contents of the array using a for loop
- **Raise your hand if you have finished (or if you are stuck)**
- **Something to try for the weekend**: output the contents of the array in reverse using a for loop

Associative Arrays

- Keys can also be strings!
  - Allows you to label values more accurately
  - No implied order

Jer	Lance	92	94	100	103
'fName'	'lName'	'HW 1'	'HW 2'	'HW 3'	'Q 2'

- Most interactions work the same

```
echo $arr['fName'];  // Outputs: 'Jer'
echo $arr['HW 2'];  // Outputs: 94
$arr['Q 2'] = 88;  // Places 88 into the $arr['Q 2'] element
```

- Some things, though, are different

```
$arr = array('fName'=>'Jer', 'lName'=>'Lance', …, 'Q 2'=>103);
$arr['Q 1'] = 22;
array_push($arr, 'Q 3'=>99);
```

Associative Arrays

Wait a minute....

HOW DO WE LOOP THROUGH THIS?

Jer	Lance	92	94	100	103
'fName'	'lName'	'HW 1'	'HW 2'	'HW 3'	'Q 2'

- We have a new loop (sort of)...the **foreach( )** loop

```
foreach($arrayName as $key=>$value) {


}
```

foreach( )

- How it works...
  - PHP creates an ***iterator*** which is really just a fancy term for a pointer
  - The ***foreach*** bit says *"one at a time, point to each member of the object I'm asking about "*
    - In this case, an array called *$arrayName*
  - The ***as*** bit says *"at each stop, store the key of the element in $key and the value in $value"*
    - These are variables...they can be anything
    - I often use $k=>$v (because I'm lazy), or something related to the data that they will hold

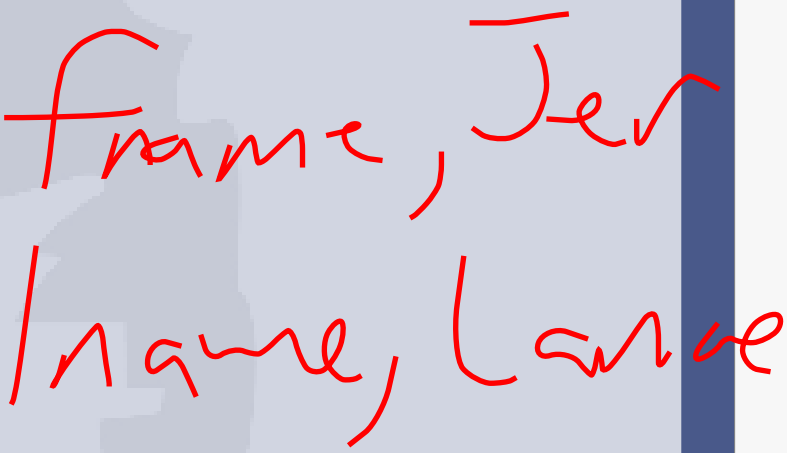
```
foreach($arrayName as $key=>$value) {


}
```

foreach( )

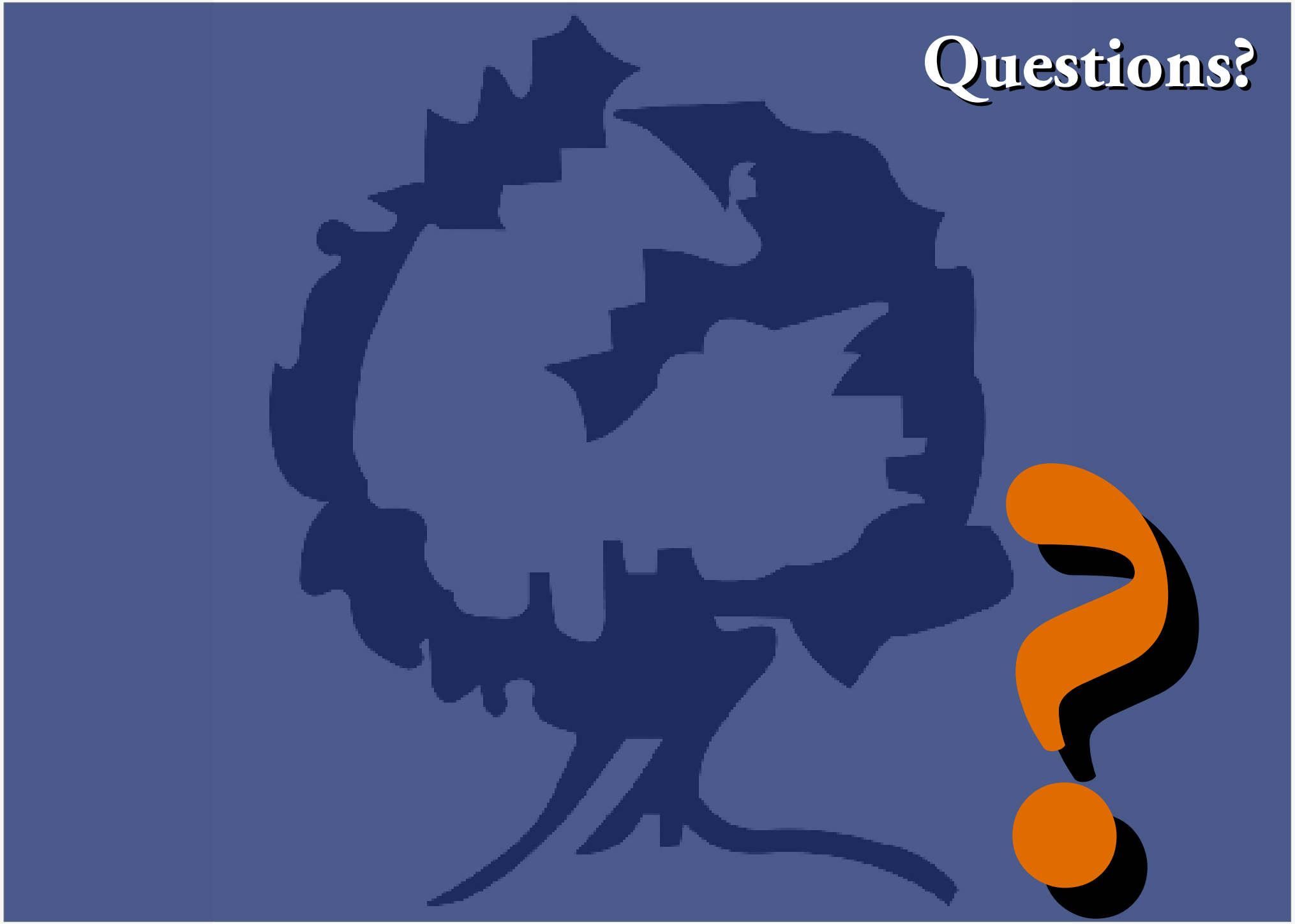
Jer	Lance	92	94	100	103
'fName'	'lName'	'HW 1'	'HW 2'	'HW 3'	'Q 2'

- An example:

```
foreach($arrayName as $k=>$v) {
        echo "$k, $v<br>";
}
```



Questions?



Let's Try It!

- **Manipulating associative arrays**
- Create a basic page ***InClass-AssocArray.php***
  - Create an array ***$myCar*** with values 'Geo', 'Metro', 'red', 100.00, 'poor', and '123456'. The keys for those values should be 'Make', 'Model', 'Color', 'Value', 'Condition', and 'ID' respectively.
    - Output the original array using a foreach( )
  - Change the condition to 'very poor' and add an element for 'Owner' with the value 'Jer'
- **Raise your hand if you have finished (or if you are stuck)**

Multidimensional Arrays

- Arrays can hold pretty much any data type
  - This includes other arrays

```
$arr1 = array('a', 'b', 'c');
$arr2 = array('d', 'e', 'f');
$arr3 = array($arr1, $arr2);
```

  - What is inside of $arr3?

2D Arrays

- An array of arrays is a 2-dimensional array

```
$arr1 = array('a', 'b', 'c');
$arr2 = array('d', 'e', 'f');
$arr3 = array($arr1, $arr2);
```

$arr3[0]	$arr1	a	b	c
$arr3[1]	$arr2	d	e	f
		0	1	2

- Accessing the elements:

```
echo $arr3[0]; // Outputs: 'Array' (because it's $arr1)
```

- Fake algebra...
  - We know that **$arr3[0]** is really **$arr1**
    a. So $arr3[0] = $arr1 (they're the same thing)
    b. How do we view the first element of $arr1? echo $arr1[0];
    c. So $arr1[0] = $arr[0][0] (if we put [0] on one side, we do on the other)

```
echo $arr3[0][0]; // Outputs: 'a'
```

Another View of 2D Arrays

- An array of arrays is a 2-dimensional array

```
$arr1 = array('a', 'b', 'c');
$arr2 = array('d', 'e', 'f');
$arr3 = array($arr1, $arr2);
```

$array3	0	1	2
0	a	b	c
1	d	e	f

- Accessing the elements:

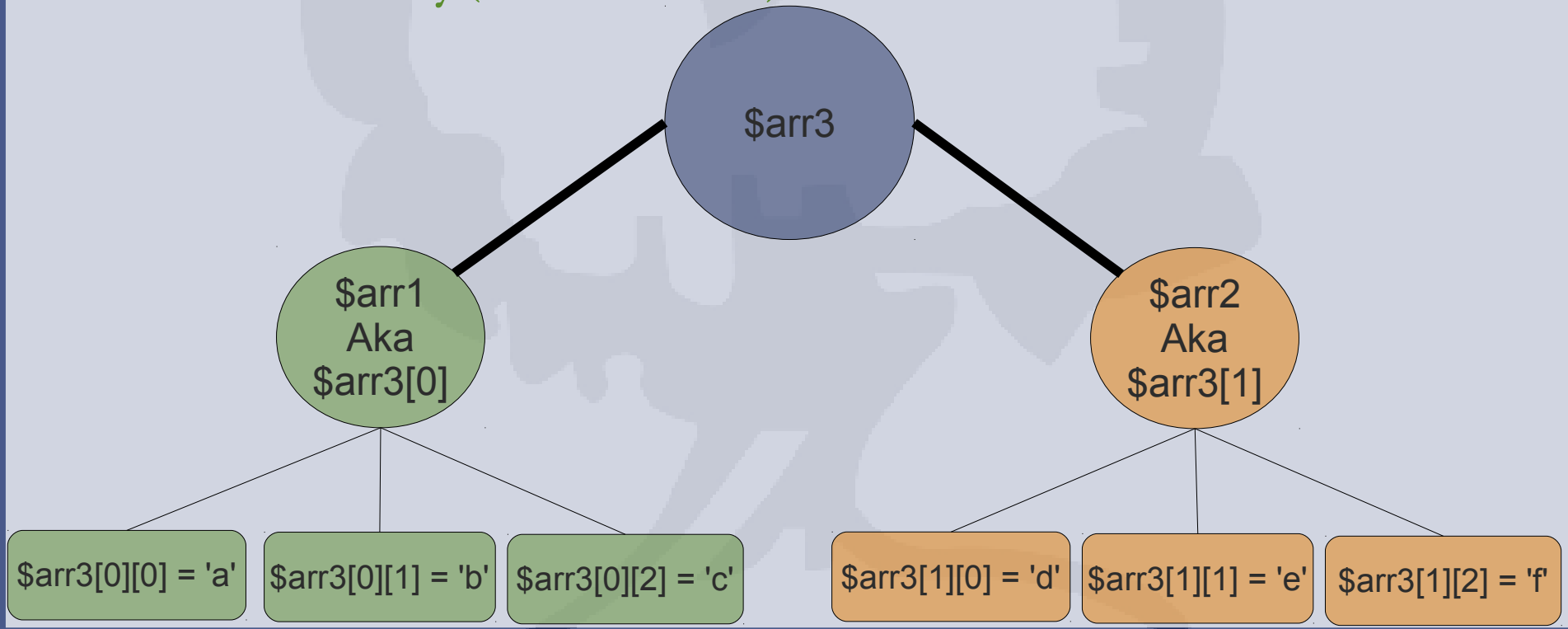
```
echo $arr3[0][0]; // Outputs: 'a' (row 0, col 0)
echo $arr3[1][2]; // Outputs: 'f' (row 1, col 2)
```

One More View of 2D Arrays

- An array of arrays is a 2-dimensional array

```
$arr1 = array('a', 'b', 'c');
$arr2 = array('d', 'e', 'f');
$arr3 = array($arr1, $arr2);
```

$array3	0	1	2
0	a	b	c
1	d	e	f



Multidimensional Arrays

- Other ways of creating multidimensional arrays
  - Declaration

```
$arr = array( array('a', 'b', 'c'), array('d', 'e', 'f'));
```

  - Pushing / Unshifting

```
$arr = array( );
array_push(array('d','e','f'));
array_unshift(array('a','b','c'));
```

$arr	0	1	2
0	a	b	c
1	d	e	f

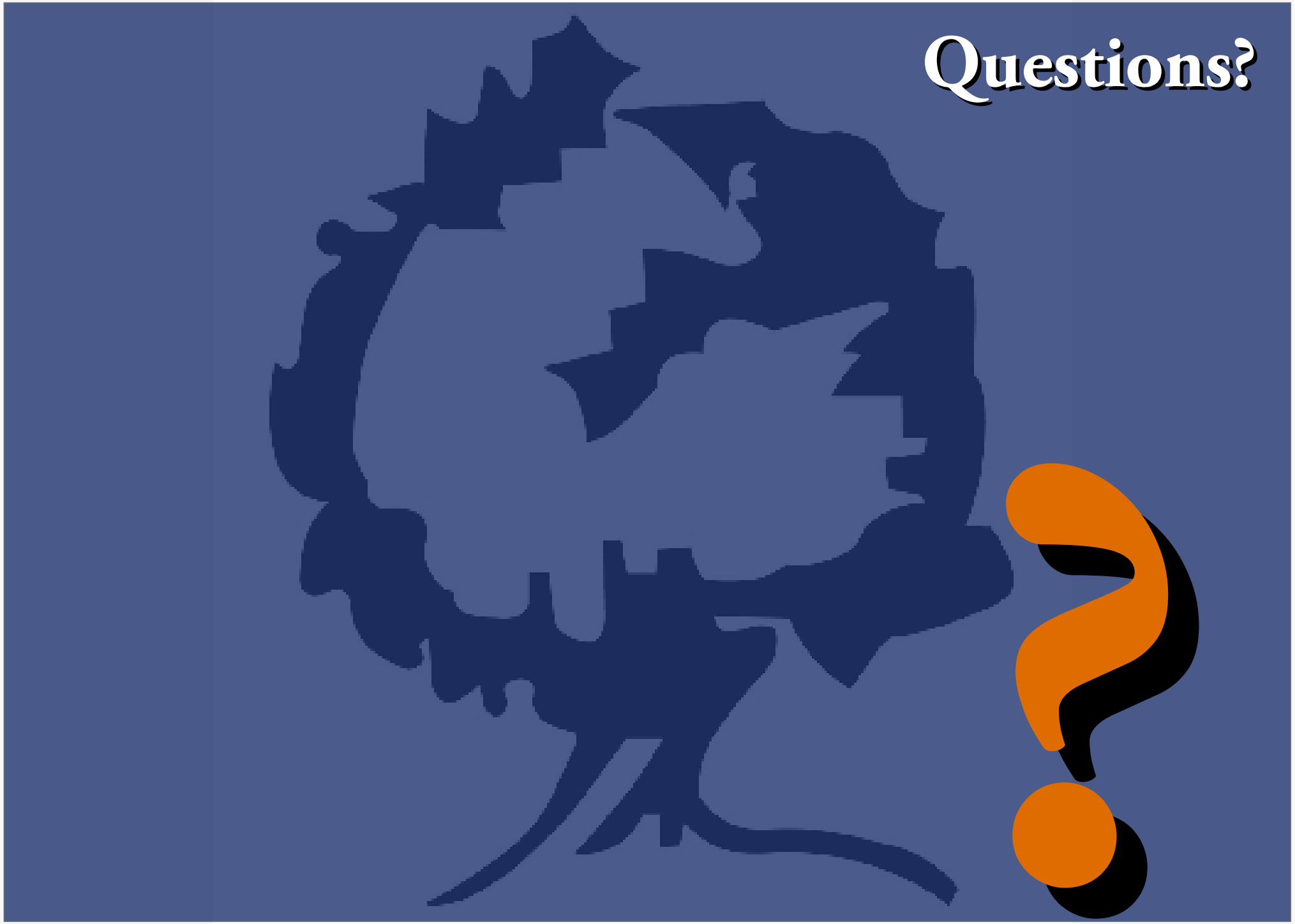
Multidimensional Arrays

- Of course, you can have 3D arrays too...

```
$arr = array(
    array(
        array('a', 'b', 'c'), array('d', 'e', 'f')
    ), array(
        array('g', 'h', 'i'), array('j', 'k', 'l')
    );
```

- It's like a cube of letters...

Questions?



Outputting 2D Arrays

- Remember those nested loops?

```
for($i = 0; $i < count($arr); ++$i) {
    for($j = 0; $j < count($arr[$i]); ++$j) {
        echo “Cell $i, $j: ”;
        echo $arr[$i][$j];
        echo '<br>';
    }
}
```

$arr	0	1	2
0	a	b	c
1	d	e	f

2D Associative Arrays

- These really come in handy when used with associative arrays

```
$student1 = array('Name'=>'Jer', 'HW'=>88, 'Proj'=>90, 'Exams'=>40);
$student2 = array('Name'=>'Sue', 'HW'=>98, 'Proj'=>105, 'Exams'=>87);
$student3 = array('Name'=>'Walt', 'HW'=>35, 'Proj'=>10, 'Exams'=>19);
$gradeBook = array('1234'=>$student1, '3452'=>$student2, '1432'=>$student3);
```

$gradeBook	Name	HW	Proj	Exams
1234	Jer	88	90	40
3452	Sue	98	105	87
1432	Walt	35	10	19

2D Associative Arrays

$gradeBook	Name	HW	Proj	Exams
1234	Jer	88	90	40
3452	Sue	98	105	87
1432	Walt	35	10	19

- Getting data out

```
echo $gradeBook['1234']['HW'];  // Outputs: 88
echo $gradeBook['1432']['Name'];  // Outputs: Walt
$gradeBook['3452']['Exams'] = 97;  // Changes Sue's Exam score to 97
```

Looping Through 2D Associative Arrays

$gradeBook	Name	HW	Proj	Exams
1234	Jer	88	90	40
3452	Sue	98	105	87
1432	Walt	35	10	19

```
foreach($gradeBook as $studID=>$studRecord) {
    echo "Student ID: $studID - ";
    foreach($studRecord as $fieldName=>$value) {
        echo "$fieldName ($value) ";
    }
    echo '<br>';
}
```

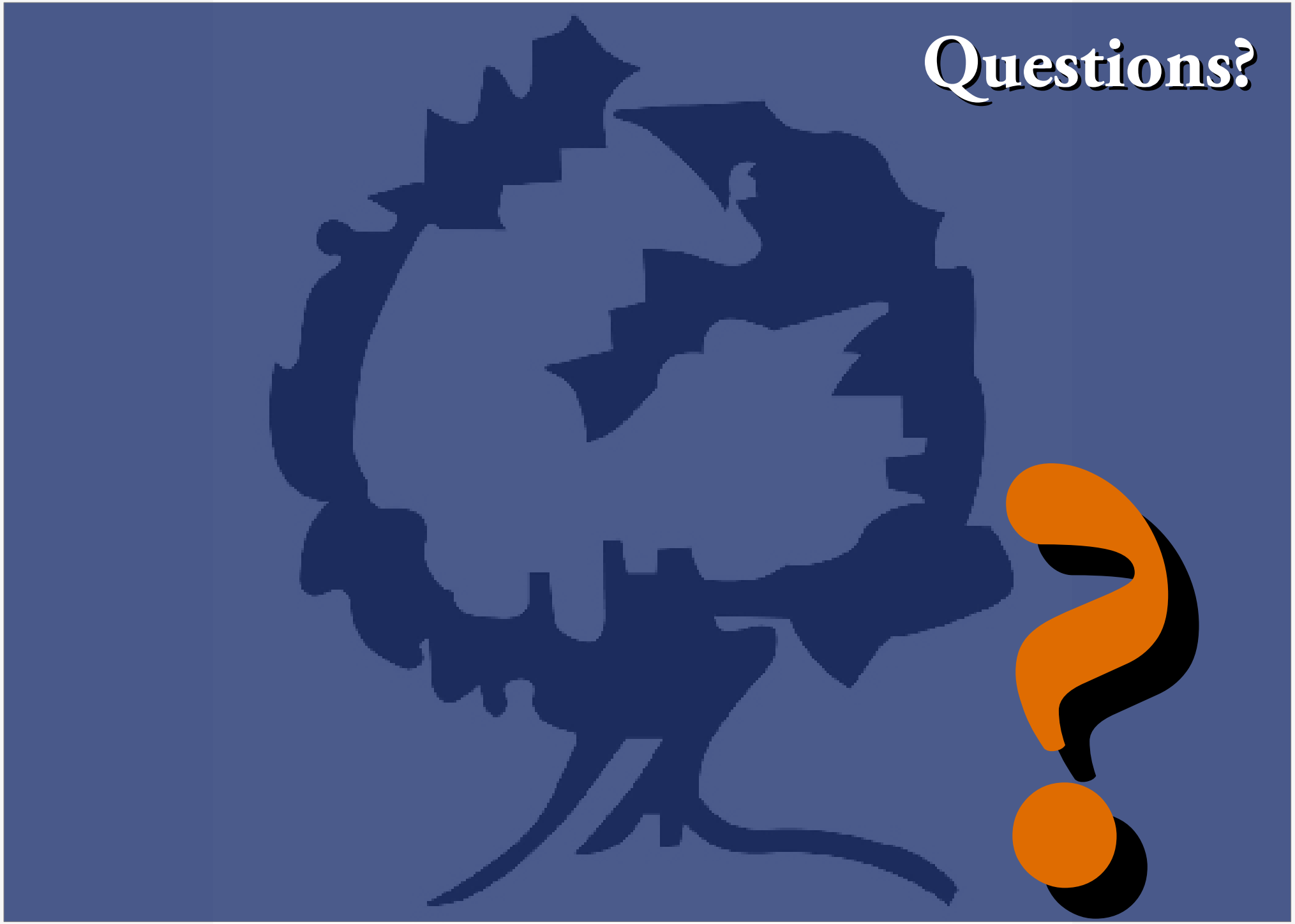
Output:

Student ID: 1234 – Name (Jer) HW (88) Proj (90) Exams (40)

Student ID: 3452 – Name (Sue) HW (98) Proj (105) Exams (87)

Student ID: 1432 – Name (Walt) HW (35) Proj (10) Exams (19)

Questions?



Let's Try It!

- **Outputting 2D Associative Arrays as Tables**
- Create a basic page ***InClass-ArrayTable.php***
  - Copy the array that I give you to use and paste it into your code
    - http://cis.hfcc.edu/~jklance/cis222/sandbox/lecture07-Arrays/bigArray.txt
  - Output the data in a table
  - **Extra hard:** Use the keys as table headers!
    - Save this for last, work on the regular output first
- **Raise your hand if you have finished (or if you are stuck)**

Next Lecture

- Arrays