



**TRY
PHP**

Level 2

Arrays

Multidimensional Arrays & Array Functions



An Array of Games

Arrays can help us organize data.

index.php

```
<?php  
  
$games = array(  
    'sorry',  
    'blackjack',  
    'poker',  
    'life',  
    'scrabble',  
);
```

sorry

blackjack

poker

life

scrabble

Groups of Games

How can we better organize our list of games?

sorry

blackjack

poker

life

scrabble



Imagining Two Groups

Splitting our games into two groups can help us sort and recall the data.

Tabletop Games

sorry

life

scrabble

Card Games

blackjack

poker



Another Dimension

index.php

```
<?php
$games = array(
    'tabletop' => 'sorry'
);
```

Tabletop Games

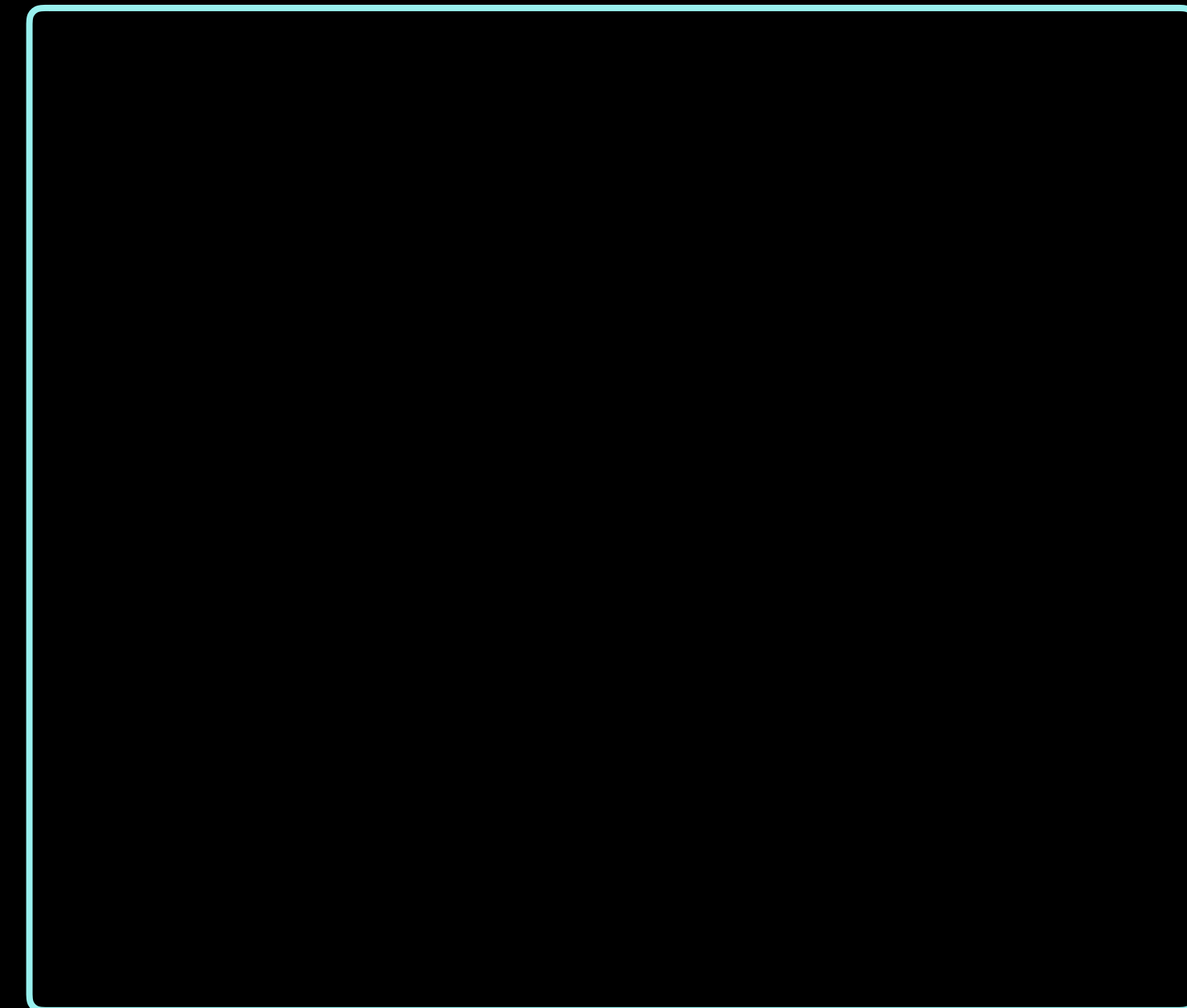
sorry

Another Dimension

index.php

```
<?php
$games = array(
    'tabletop' => array()
);
```

Tabletop Games



Another Dimension

index.php

```
<?php
$games = array(
    'tabletop' => array(
        'sorry',
        'life',
        'scrabble',
    ),
);
```

Tabletop Games

Sorry

Life

Scrabble

Another Dimension

index.php

```
<?php
$games = array(
    'tabletop' => array(
        'sorry',
        'life',
        'scrabble',
    ),
    'card' => array(
        'poker',
        'blackjack',
    ),
);
```

Tabletop Games

Sorry

Life

Scrabble

Card Games

Poker

Blackjack

Array Inspection

If we print_r our \$games array, you can see the multidimensional structure.

Output

```
<?php  
print_r($games);
```

```
Array(  
    [tabletop] => Array(  
        [0] => sorry  
        [1] => life  
        [2] => scrabble  
    )  
    [card] => Array(  
        [0] => poker  
        [1] => blackjack  
    )  
)
```



Accessing Data

By using the array's key, we can see the array value.

index.php

```
<?php
```

```
print_r($games['tabletop']);
```

Output

```
Array (  
[0] => sorry  
[1] => life  
[2] => scrabble  
)
```


Accessing Data

By using the array's key, we can see the array value.

index.php

```
<?php  
  
print_r($games['tabletop']);  
  
print_r($games['card']);
```

Output

```
Array (  
[0] => poker  
[1] => blackjack  
)
```


Accessing Data

By using the array's key, we can see the array value.

index.php

```
<?php  
  
print_r($games['tabletop']);  
  
print_r($games['card']);  
  
echo $games['tabletop'][0];
```

Output

sorry

Modifying the Data

Instead of single item access, we can use the same methods to change a value.

index.php

```
<?php

$games[ 'card' ][ 0 ] = 'rummy';

print_r( $games[ 'card' ] );
```

Output

```
Array (
  [0] => rummy
  [1] => blackjack
)
```


Array Functions: count

This function lets us count all the items in an array.

index.php

```
<?php
$people = array(
    'David',
    'Jennifer',
    'Falken',
    'Joshua',
);

echo count($people);
```

Output

4

Array Functions: implode

implode joins all elements of the array into a string.

index.php

```
<?php
$people = array(
    'David',
    'Jennifer',
    'Falken',
    'Joshua',
);

echo implode(' ', $people);
```

*the character that separates
the combined array values*

the array to combine

Output

```
'David Jennifer
Falken Joshua'
```


Array Functions: shuffle

shuffle changes the array in place to a random order.

index.php

```
<?php
$people = array(
    'David',
    'Jennifer',
    'Falken',
    'Joshua',
);

// Randomize the array.
shuffle($people);

echo implode(' ', $people);
```

Output

```
'Jennifer David
Joshua Falken'
```


Array Functions: asort

asort will sort the array values, in place, in alphabetical order.

index.php

```
<?php
$people = array('David', 'Jennifer', 'Falken', 'Joshua');

// Sort the array alphabetically.
asort($people);

echo implode(' ', $people);
```

Output

```
'David Falken
Jennifer Joshua'
```


What Have We Learned?

Let's have a quick review.

- Multidimensional array basics
- Some simple array functions
 - shuffle
 - implode
 - asort
 - count

And Now, It's Challenge Time!





**TRY
PHP**