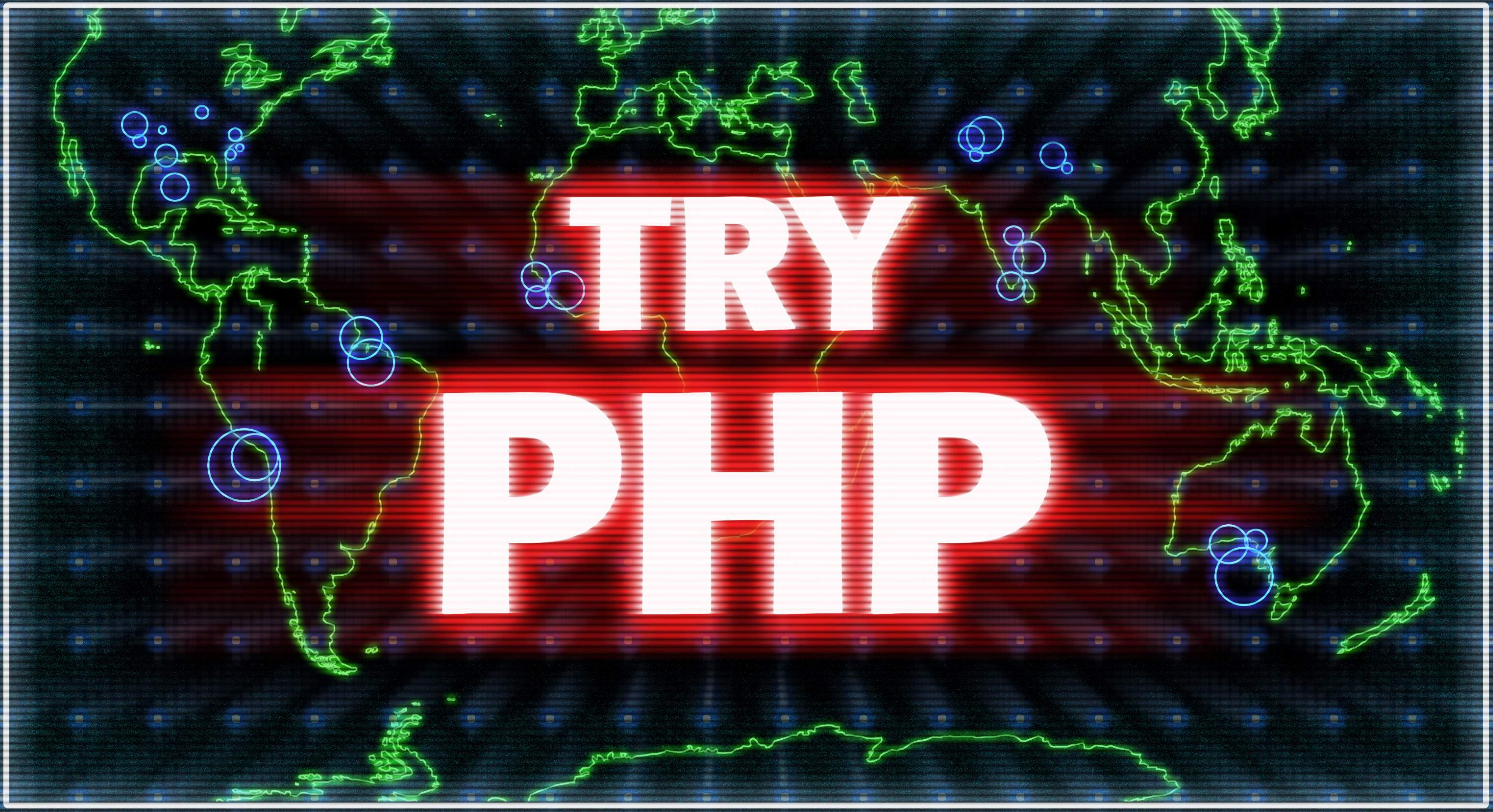


TRY PHP

Level 2

Arrays

Simple & Associative Arrays

Why an Array?

Variables alone will not scale. We need a better way to keep our data.

index.php

```
<?php
// We could keep going with variables
$meteor_1 = 'Hoba';
$meteor_2 = 'Cape York';
$meteor_3 = 'Campo del Cielo';
$meteor_4 = 'Canyon Diablo';
…
$meteor_42 = 'Prefect';
```

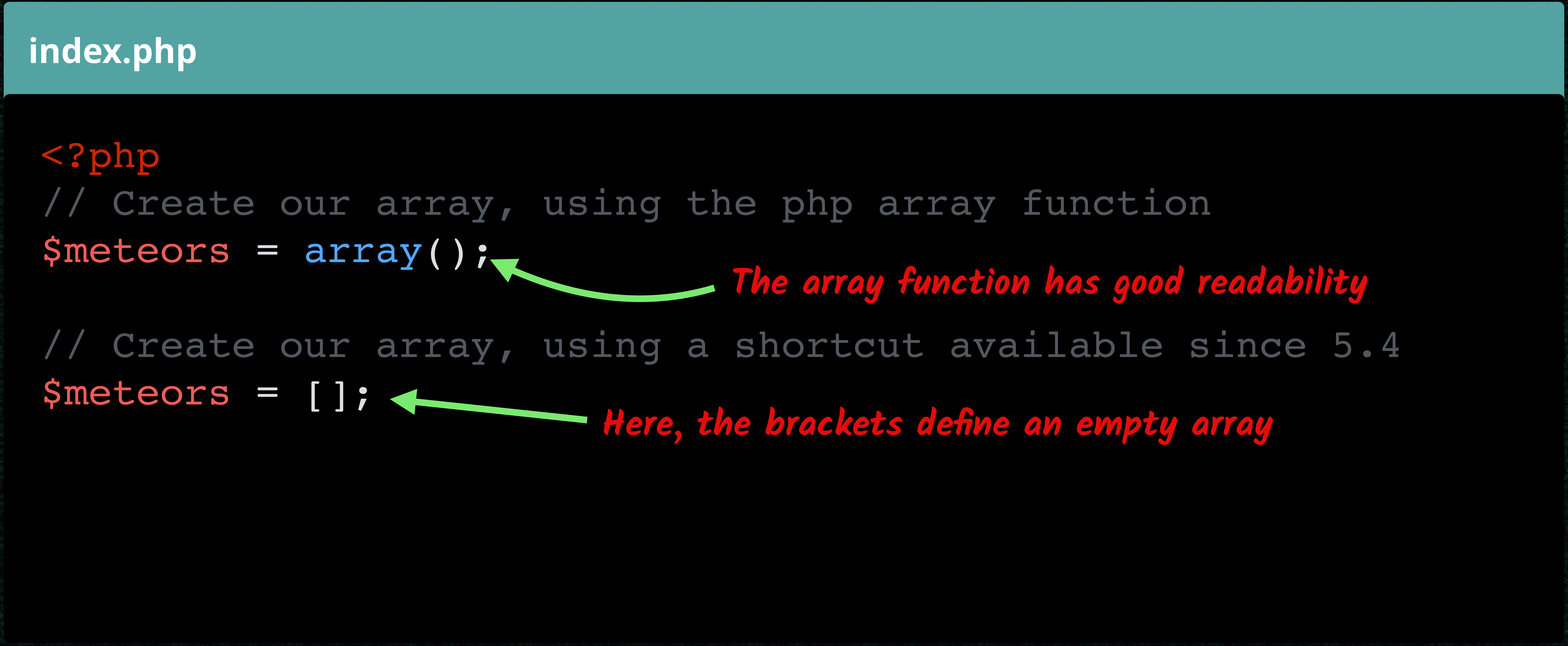
Arrays, a Map

An array maps values to keys, like an address for setting and recalling.

Key	Value
0	Hoba
1	Cape York
2	Campo del Cielo
3	Canyon Diablo

Creating an Array

Let’s create an empty array to hold our meteorite data.



index.php

```php
<?php
// Create our array, using the php array function
$meteors = array();

// Create our array, using a shortcut available since 5.4
$meteors = [];
```

The array function has good readability

Here, the brackets define an empty array

Array With Values

We can create an array with one or more key value pairs using the same function.

index.php

```php
<?php
// Create our array with a single value
$meteors = array('Hoba');
$meteors = ['Hoba'];

// Create array with multiple values
$meteors = array('Hoba','Cape York');

// Echo the array
echo($meteors);
```

echo will not show the data within the array

Output

```
Array
```

Array With Values

We can create an array with one or more key value pairs using the same function.

index.php

```php
<?php
// Create our array with a single value
$meteors = array('Hoba');
$meteors = ['Hoba'];

// Create array with multiple values
$meteors = array('Hoba','Cape York');

// Let's take a look at our array
with an internal function
print_r($meteors);
```

print_r will echo human-readable output

Output

```
Array (
[0] => Hoba
[1] => Cape York
)
```

Adding More Data to Our Array

We can append new values by placing square brackets after the array variable.

index.php

```php
<?php
…
// Let's add two more items
$meteors[] = 'Campo del Cielo';
$meteors[] = 'Canyon Diablo';

print_r($meteors);
```

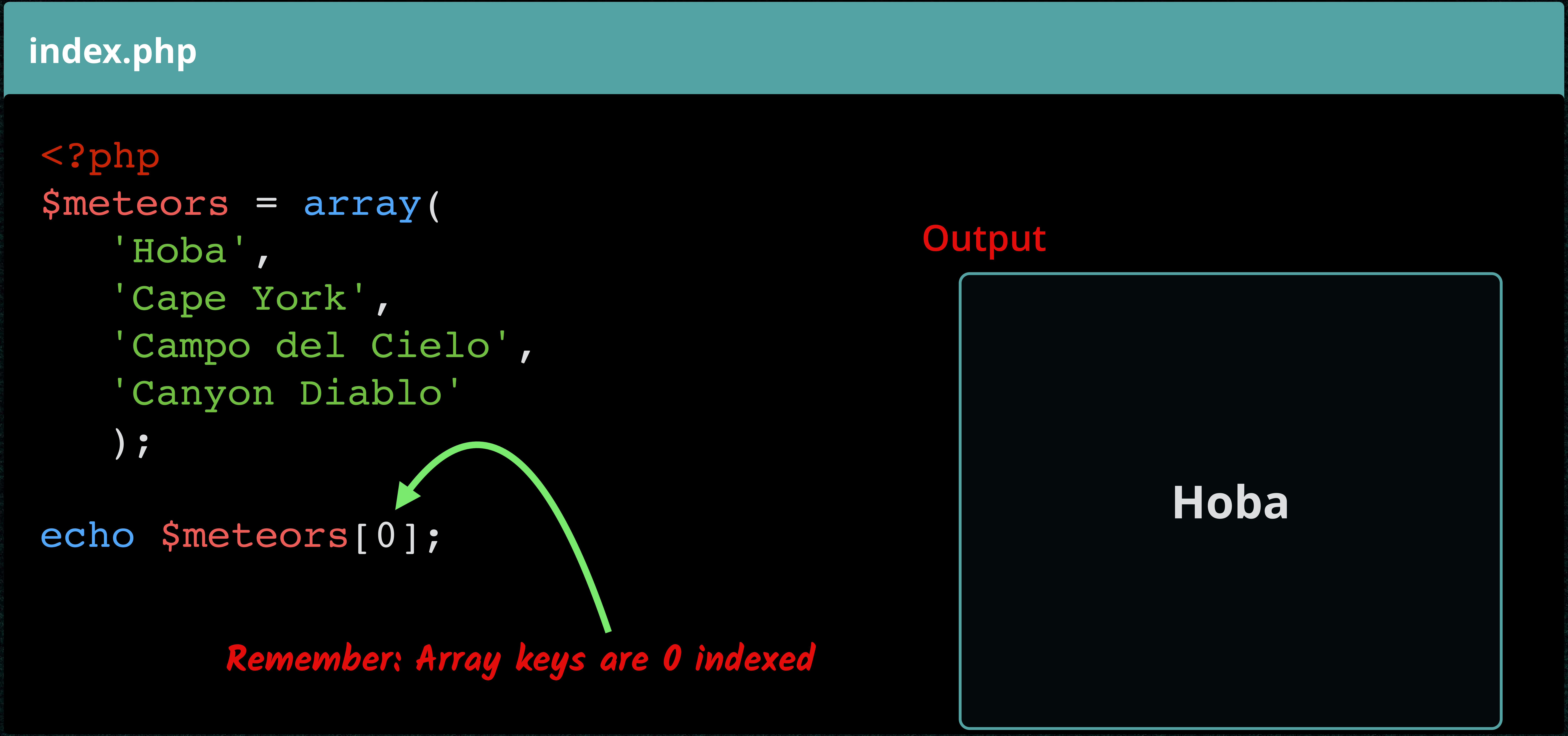
Empty brackets after the variable name indicate a new item in the array

Output

```
Array (
[0] => Hoba
[1] => Cape York
[2] => Campo del Cielo
[3] => Canyon Diablo
)
```

How Can We Access This Data?

Placing the key, or index, inside the square bracket gives us access to the value.



index.php

```
<?php
$meteors = array(
   'Hoba',
   'Cape York',
   'Campo del Cielo',
   'Canyon Diablo'
   );

echo $meteors[0];
```

Remember: Array keys are 0 indexed

Output

```
Hoba
```

How Can We Access This Data?

Placing the key, or index, inside the square bracket gives us access to the value.

index.php

```php
<?php
$meteors = array(
    'Hoba',
    'Cape York',
    'Campo del Cielo',
    'Canyon Diablo'
    );

echo $meteors[0];
echo $meteors[1];
echo $meteors[3];
```

Output

```
Hoba
Cape York
Canyon Diablo
```

Modifying an Existing Item

Placing the key inside also allows us access to modify the value.

index.php

```php
<?php
...
$meteors[0] = 'Los Angeles';

print_r($meteors);
```

Choose your key to modify

Then modify the value

Output

```
Array (
[0] => Los Angeles
[1] => Cape York
[2] => Campo del Cielo
[3] => Canyon Diablo
)
```

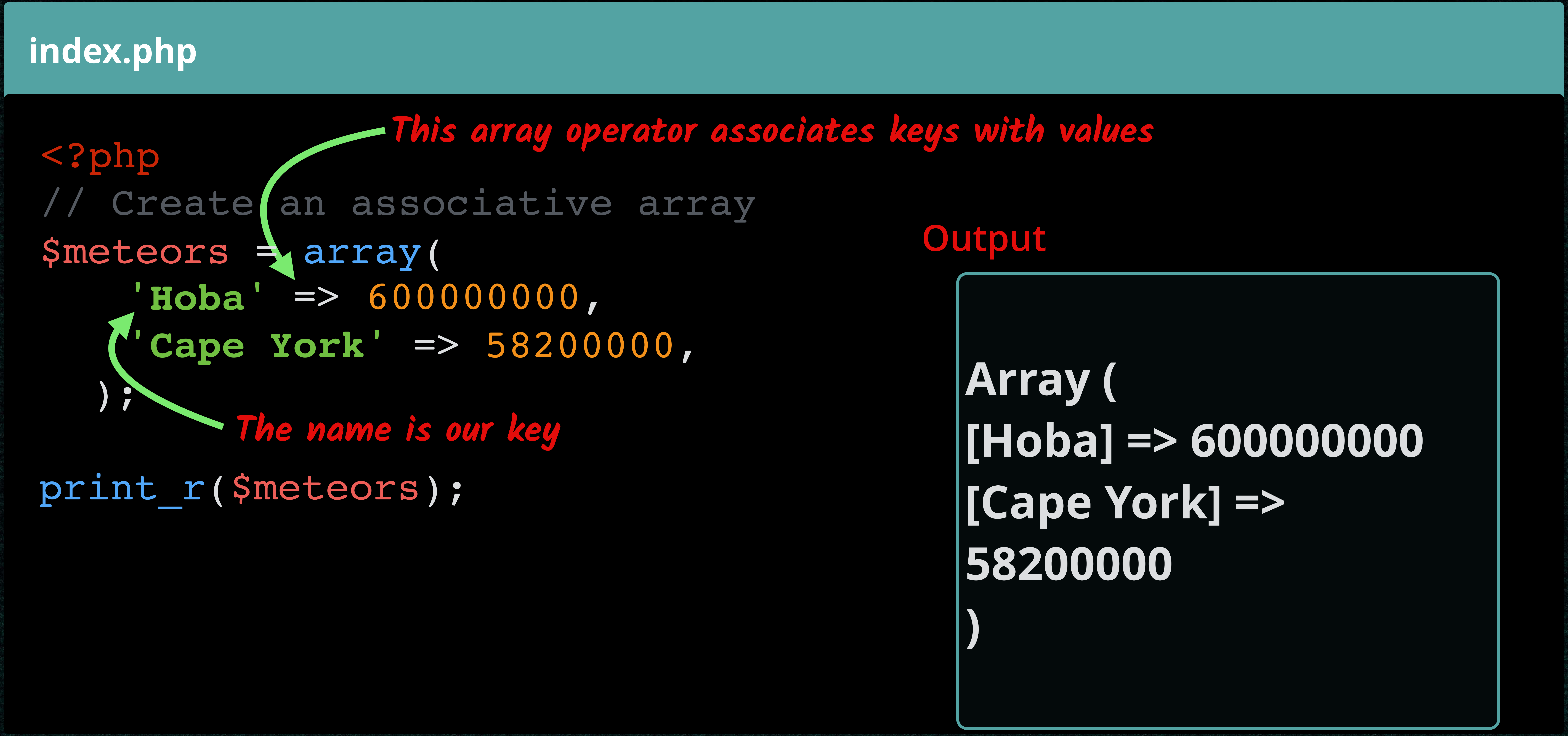
Storing Even More Data in an Array

What if we want to store more information about the meteorite?

Name	Weight	Location	Year
Hoba	600000000	-19.58333, 17.91667	1920
Cape York	58200000	76.13333, -64.93333	1818
Campo del Cielo	50000000	-27.46667, -60.58333	1576
Canyon Diablo	30000000	35.05, -111.03333	1891

Associative vs. Index Arrays

Associative arrays allow us to use strings as the key.



index.php

```php
<?php
// Create an associative array
$meteors = array(
    'Hoba' => 600000000,
    'Cape York' => 58200000,
  );

print_r($meteors);
```

This array operator associates keys with values

The name is our key

Output

```
Array (
[Hoba] => 600000000
[Cape York] =>
58200000
)
```

Accessing an Item in the Array

Instead of the numerical index, we now use the string key for access.

index.php

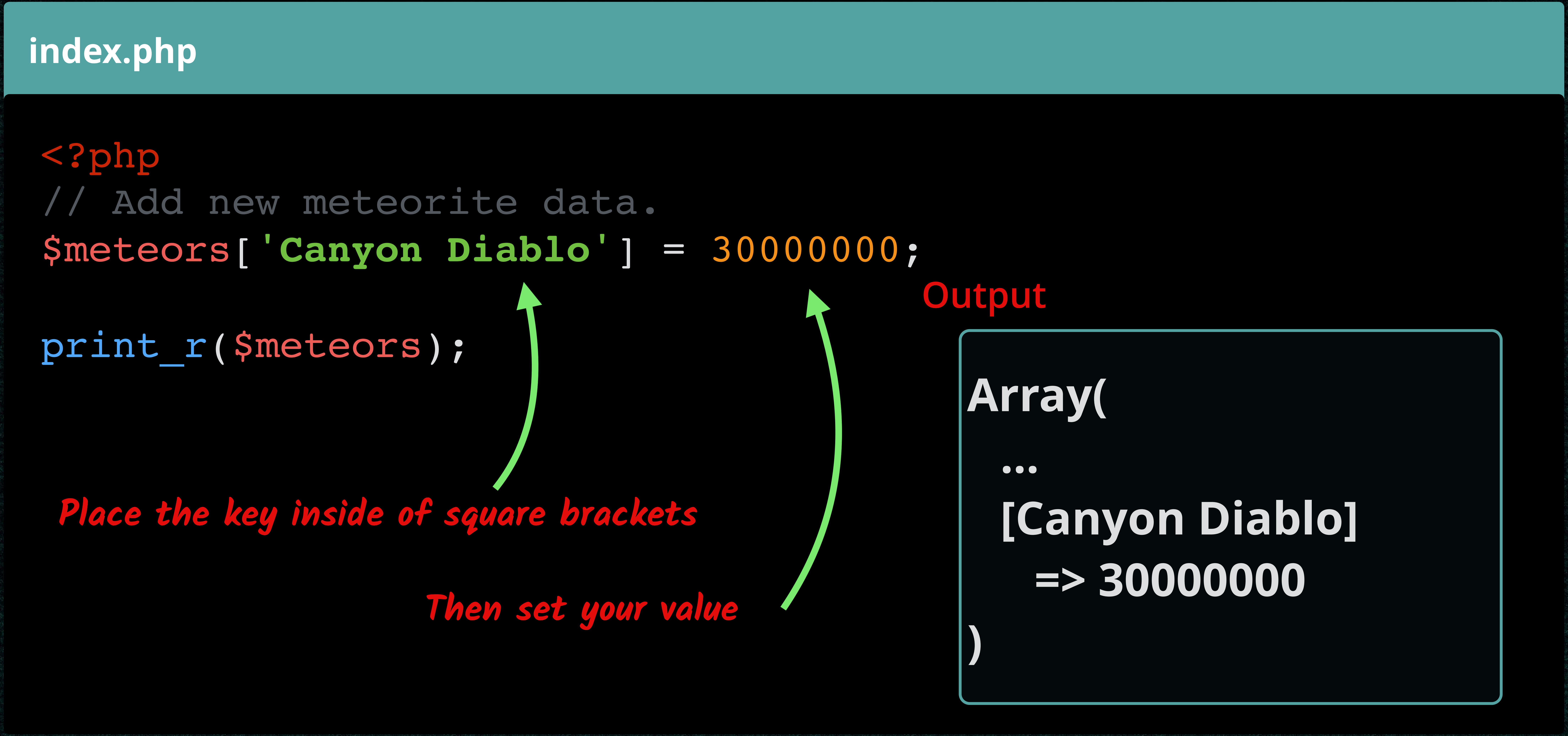
```php
<?php

// Access our data.
echo $meteors['Hoba'];
echo $meteors['Cape York'];
```

Output

```
600000000
58020000
```

Appending a New Item

Using a string key, we can add a new item as well.



index.php

```php
<?php
// Add new meteorite data.
$meteors['Canyon Diablo'] = 30000000;

print_r($meteors);
```

Place the key inside of square brackets

Then set your value

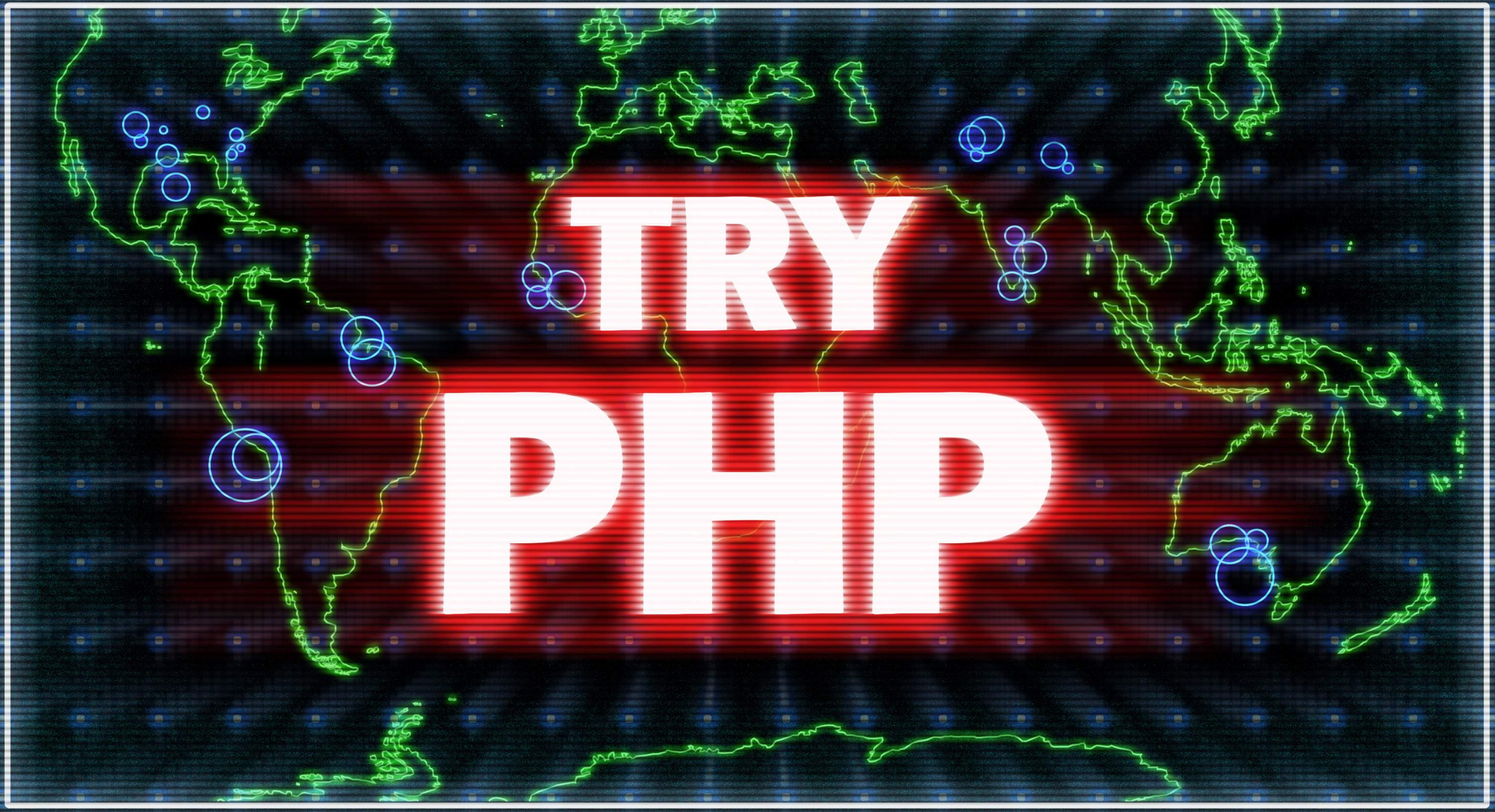
Output

```
Array(
  ...
  [Canyon Diablo]
    => 30000000
)
```

What Have We Learned?

Let’s have a quick review.

- Numerical indexed arrays
- Associative arrays
- Array creation with values
- Modification of array data


TRY
PHP