MEC ACADEMY

PHP Development Course

MEC Academy

Agenda

- Loops
  - For loop
  - While loop
  - Do while loop
  - Foreach loop
- Arrays
  - Indexing array
  - Associative array
  - Multidimensional array
- Array Operators
- Array functions
- Functions
  - Built in function
  - User defined function

MEC ACADEMY

LOOPS

Arithmetic

MEC ACADEMY

When to use loops?

- Often when you write code, you want the same block of code to run repeatedly a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops.
- Loops are used to execute the same block of code again and again, **if a certain condition is true.**
- There are 4 types of loop: **for, while, do while, foreach**

For Loop

Arithmetic

MEC ACADEMY

For Loop

- PHP for loop can be used to traverse set of code for the specified number of times.

- It should be used if the number of iterations is known otherwise use while loop. This means for loop is used when you already know how many times you want to execute a block of code.

MEC ACADEMY

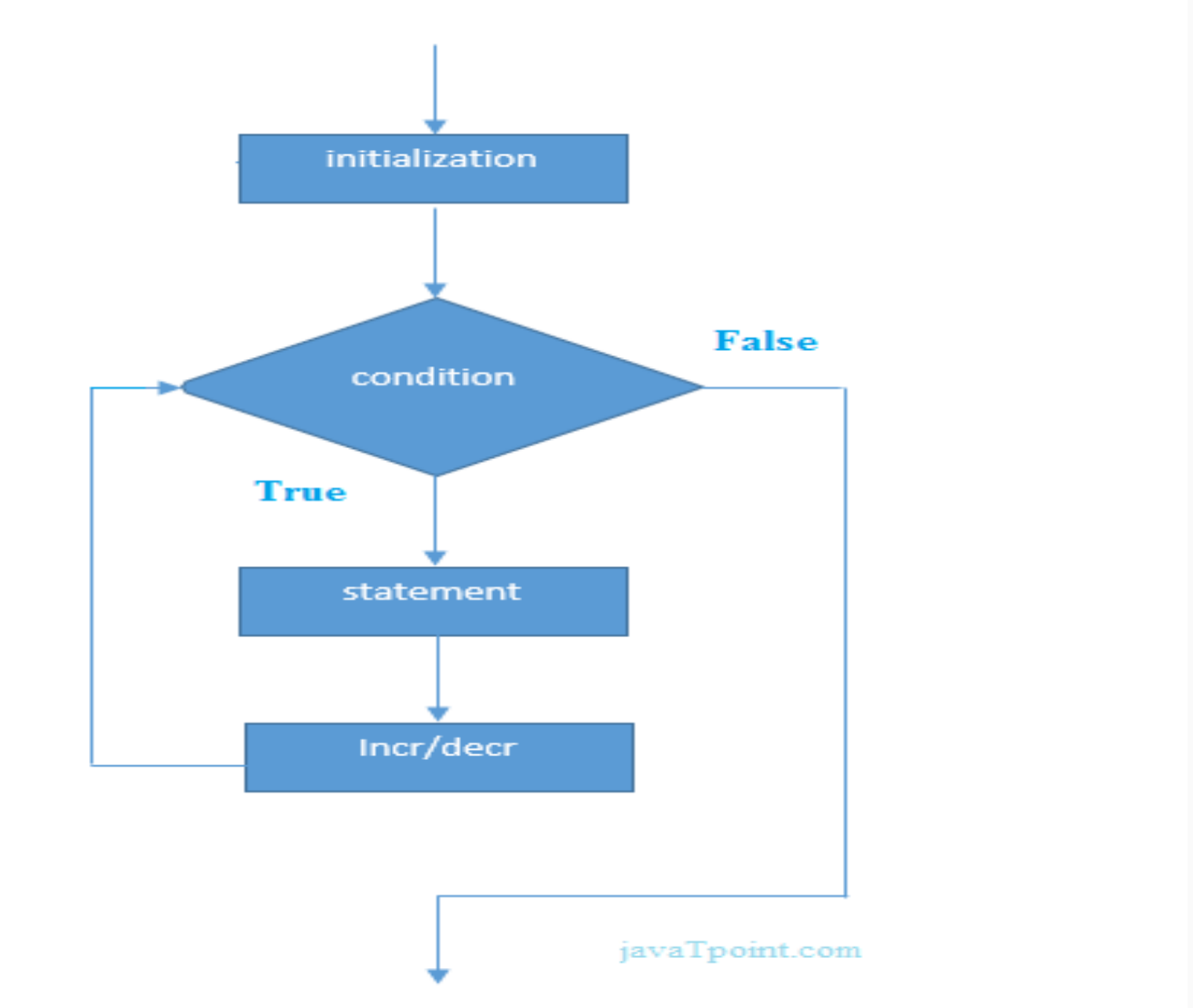
For Loop Syntax

```
for( expression1 ; condition ; expression2){
//code to be executed in each iteration
}
```

- **expression1** is executed once at the start. Here, you usually set the initial value of a counter.
- **The condition** expression is tested before each iteration. If the expression returns false, iteration stops. Here, you usually test the counter against a limit.
- **expression2** is executed at the end of each iteration. Here, you usually adjust the value of the counter(increment or decrement the counter).

MEC ACADEMY

For Loop


initialization
condition
False
True
statement
Incr/decr
javaTpoint.com

While Loop

Arithmetic

MEC ACADEMY

While Loop

- PHP while loop can be used to traverse set of code like for loop. The while loop executes a block of code repeatedly until the condition is **FALSE**. Once the condition gets **FALSE**, it exits from the body of loop.

- It should be used if the number of iterations is not known.

MEC ACADEMY

While Loop Syntax

```
while( condition){
//code to be executed
}
```

- **The condition** expression is tested before each iteration. If the expression returns false, iteration stops.
- **Example**

```
$x = 0;
while($x <= 100) {
    echo " <br> The number is: $x ";
    $x+=10;
}
```

MEC ACADEMY

Do While Loop

Arithmetic

MEC ACADEMY

Do While Loop

- do-while loop can be used to traverse set of code like while loop. But The do-while loop is guaranteed to run at least once.
- The do-while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.
- The do-while loop is very much like the while loop except the condition check. The main difference between both loops is that while loop checks the condition at the beginning, whereas do-while loop checks the condition at the end of the loop.

MEC ACADEMY

Do While Loop Syntax

```
do{
//code to be executed
}
while( condition);
```

- **The condition** expression is tested before each iteration. If the expression returns false, iteration stops.
- **Example**

```
$x = 100000;
do {
  echo " <br> The number is: $x ";
   $x++;
} while ($x <= 5);
```

MEC ACADEMY

Difference between while and do-while loop

while Loop	do-while loop
The while loop is also named as **entry control loop**.	The do-while loop is also named as **exit control loop**.
The body of the loop does not execute if the condition is false.	The body of the loop executes at least once, even if the condition is false.
Condition checks first, and then block of statements executes.	Block of statements executes first and then condition checks.
This loop does not use a semicolon to terminate the loop.	Do-while loop use semicolon to terminate the loop.

Foreach Loop

Arithmetic

MEC ACADEMY

Foreach Loop

- The foreach loop is used to traverse the array elements. It works only on array and object. It will issue an error if you try to use it with the variables of different datatype.

- The foreach loop works on elements basis rather than index. It provides an easiest way to iterate the elements of an array.

MEC ACADEMY

Foreach Loop Syntax

```
foreach ($array as $value) {
//code to be executed;
}
```

Or if array is associative array

```
foreach ($array as $key => $value) {
//code to be executed;
}
```

MEC ACADEMY

Flow control Keywords

Arithmetic

MEC ACADEMY

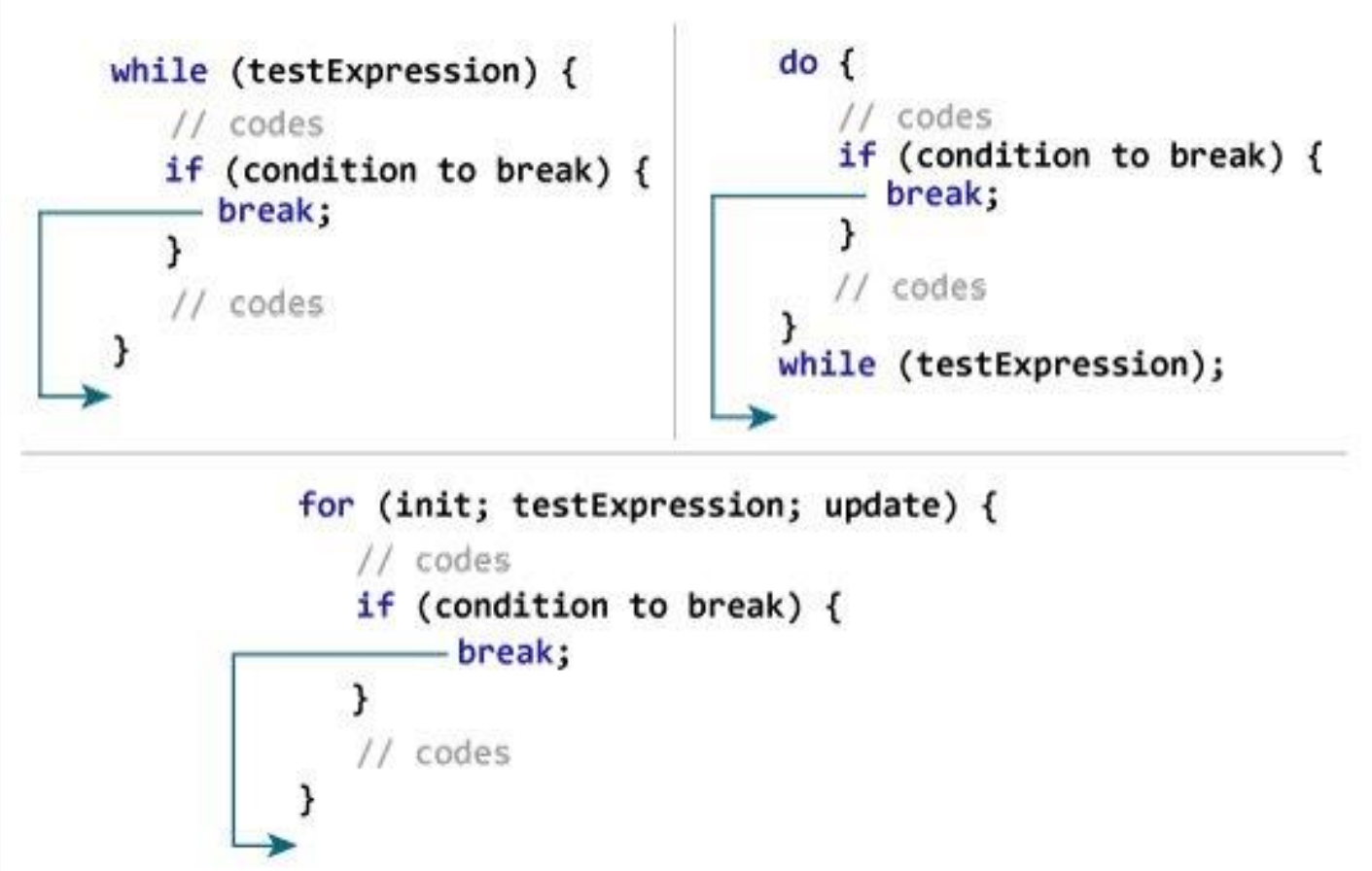
Break

Use the break statement in a loop, execution of the script will used to **jump out** of a loop.

```
for ($i = 0; $i < 5; ++$i) {
if ($i == 3)
break;
print "$i <br>";
}
```

MEC ACADEMY

How break statement works?



MEC ACADEMY

continue

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

```
for ($i = 0; $i < 5; ++$i) {
if ($i == 3)
continue;
print "$i <br>";
}
```

How continue statement works?

```
while (testExpression) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

```
do {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
} while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

MEC ACADEMY

Arrays

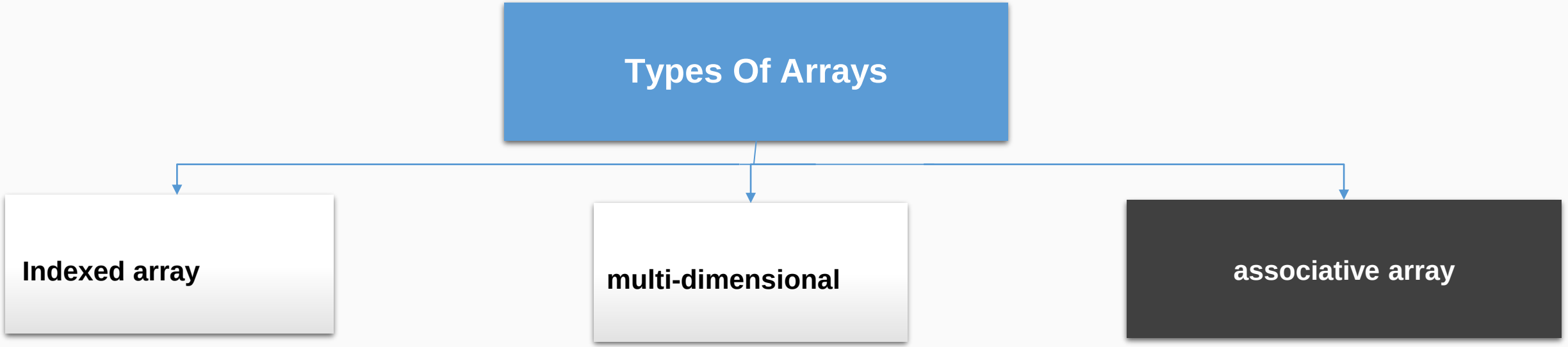
Arithmetic

MEC ACADEMY

Arrays

Arrays are an essential part of any programming language, and PHP is no exception. In PHP, arrays are a way to store multiple values in a single variable. They can be used to represent lists, tables, or even sets of data.

MEC ACADEMY


Types Of Arrays
Indexed array
multi-dimensional
associative array

Indexed Array

- Declaring Arrays in PHP In PHP, you can declare an array using the array() function. The general syntax for declaring an array is:

```
$array_name = array(value1, value2, value3,...);
```

```
$fruits = array("apple", "banana", "orange", "grape");
```

- The indices of numerically indexed arrays in PHP, start at zero by default, although you can alter this value.

MEC ACADEMY

Indexed Array

You can also create an empty array and add items to it later:

```php
$fruits = array();
$fruits[] = "apple";
$fruits[] = "banana";
$fruits[] = "orange";
$fruits[] = "grape";
```

MEC ACADEMY

Indexed Array

Accessing Array Elements PHP arrays are zero-indexed, which means the first element has an index of 0, the second element has an index of 1, and so on. **To access an element of an array,** you use square brackets [] with the index of the element you want to access:

```php
$fruits = array("apple", "banana", "orange", "grape");
echo $fruits[0]; // Output: apple
echo $fruits[1]; // Output: banana
echo $fruits[2]; // Output: orange
echo $fruits[3]; // Output: grape
```

MEC ACADEMY

Indexed Array

You can also use loops to iterate over the elements of an array:

```php
$fruits = array("apple", "banana", "orange", "grape");
foreach ($fruits as $fruit) {
    echo $fruit . "\n";
}
```

MEC ACADEMY

Multi-Dimensional Arrays

```php
$students = array(
    array("John", "Doe", 20),
    array("Jane", "Doe", 22),
    array("Bob", "Smith", 21)
);
```

To access an element of a multi-dimensional array, you use multiple indexes:

```php
echo $students[0][0]; // Output: John
echo $students[1][2]; // Output: 22
```

Hint: Use loops to access two dimensional array

MEC ACADEMY

Arrays

- To add new element in the end of array:-

  $arrName[] = newElementValue;

- To add new element in specific index in array:

  $arrName[index] = newElementValue;

- To change the value of element in specific index in array:

  $arrName[index] = newValue;

For example:-

```
$arr = [1,2,3];
$arr[] = 6; // [1,2,3,6]
$arr[4] = 10; // [1,2, 3, 6, 10]
$arr[2] = 50; // [1, 2, 50, 6, 10]
```

MEC ACADEMY

Associative Array

- To add new element in the array:-

  $arrName[“newKey”] = newElementValue;

- To change the value of element in specific index in array:

  $arrName[key] = newValue;

For example:-

```
$user =["name"=> "eman", "email"=>"eman@gmail.com"];
$user["password"] ="1425";
 // =["name"=> "eman", "email"=>"eman@gmail.com","password"=> "1425"]
$user["name"] = "eman mohamed" // edit the value of key name
// =["name"=> "eman mohamed", "email"=>"eman@gmail.com","password"=> "1425"]
```

MEC ACADEMY

Array Operators

Arithmetic

MEC ACADEMY

Array Operators

- array operators are used to compare arrays.

Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y
==	Equality	$x == $y	Returns true if $x and $y have the same key/value pairs
===	Identity	$x === $y	Returns true if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	$x != $y	Returns true if $x is not equal to $y
<>	Inequality	$x <> $y	Returns true if $x is not equal to $y
!==	Non-identity	$x !== $y	Returns true if $x is not identical to $y

MEC ACADEMY

Array Functions

Arithmetic

Sorting Array

- sort($arr) - sort arrays in ascending order
- rsort($arr) - sort arrays in descending order
- asort($arr) - sort associative arrays in ascending order, according to the value.
- arsort($arr) - sort associative arrays in descending order, according to the value.
- ksort($arr) - sort associative arrays in ascending order, according to the key.
- krsort($s) - sort associative arrays in descending order, according to the key.

❖ **Those functions take the array as a parameter and edit on the origin array.**

Note: in sorting number less than character, capital letters less than lowercase letters

MEC ACADEMY

Array functions

- array_push($arr, $newElement) – add one new element to the end of an array
- array_unshift($arr, $newElement) – add one new element to the start of an array.
- array_pop($arr) - removes and returns one element from the end of an array.
- array_shift($arr) - removes and returns one element from the start of an array.
- unset($arr[index]) – removes element in specific index or key in an array.
- array_merge($arr1, $arr2) – merges two arrays and return new array.

MEC ACADEMY

Functions

Arithmetic

MEC ACADEMY

Functions

In general: is a way to perform some task.

In PHP: Functions in PHP are blocks of code that can be defined once and called multiple times throughout a script. They allow you to encapsulate code into reusable modules, making your code more modular and easier to maintain..

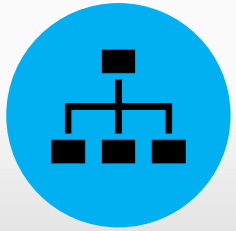
Importance of functions

Reusability

Time savers and help us to reuse code without retyping the code. Write one use many!

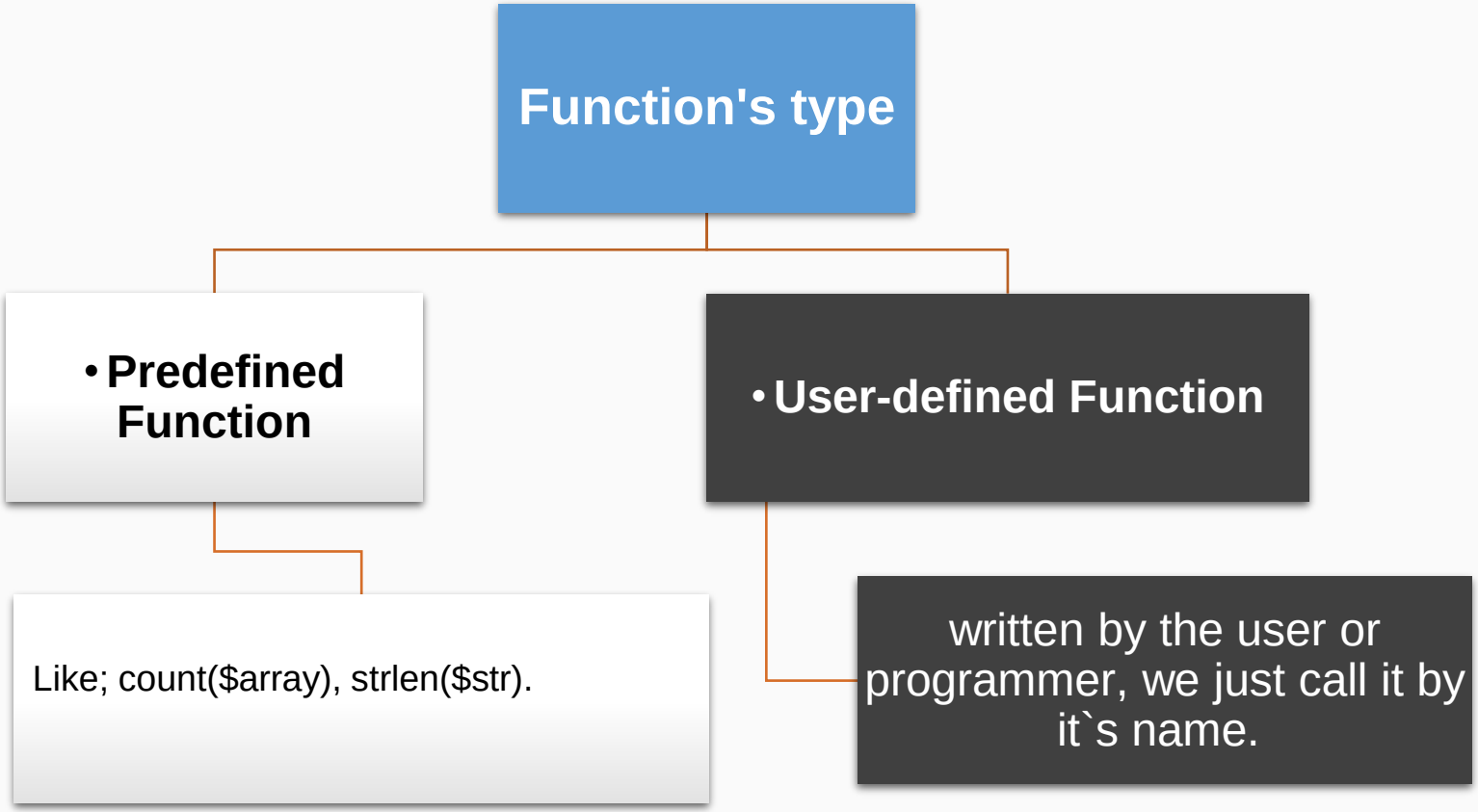
Maintenance

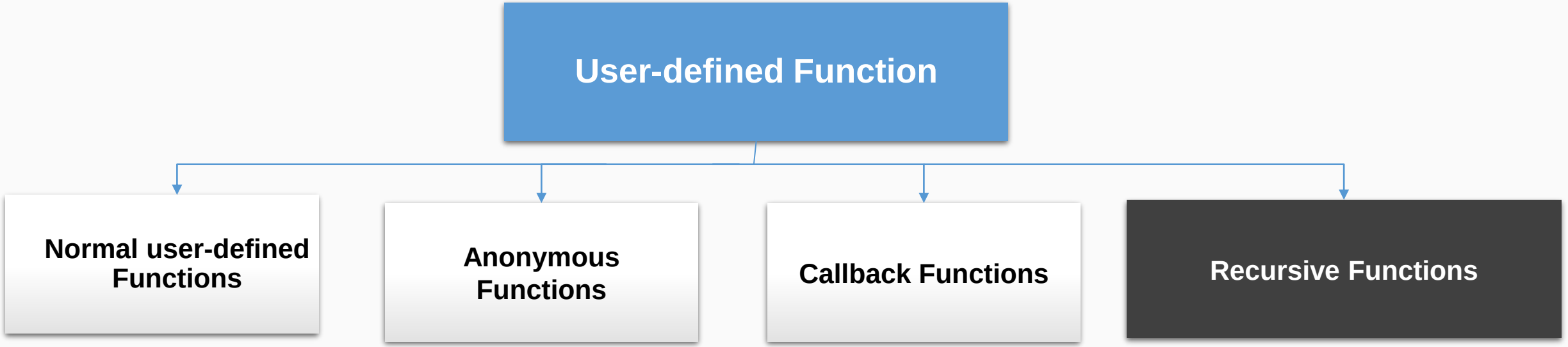
Make the maintenance process easier.



Manageable

It divides the code to tasks so that make it easier to control.


Function's type
• Predefined Function
• User-defined Function
Like; count($array), strlen($str).
written by the user or programmer, we just call it by it`s name.


User-defined Function
Normal user-defined Functions
Anonymous Functions
Callback Functions
Recursive Functions

Normal User-defined Function

These are functions that you create yourself using the "function" keyword. You can define your own parameters and return values for these functions.

```php
function add($num1, $num2) {
  $sum = $num1 + $num2;
  return $sum;
}
```

parameters

```php
$result = add(5, 10);
echo $result; // Outputs 15
```

arguments

//Call the function by its name!

What happened in memory when I call function?

MEC ACADEMY

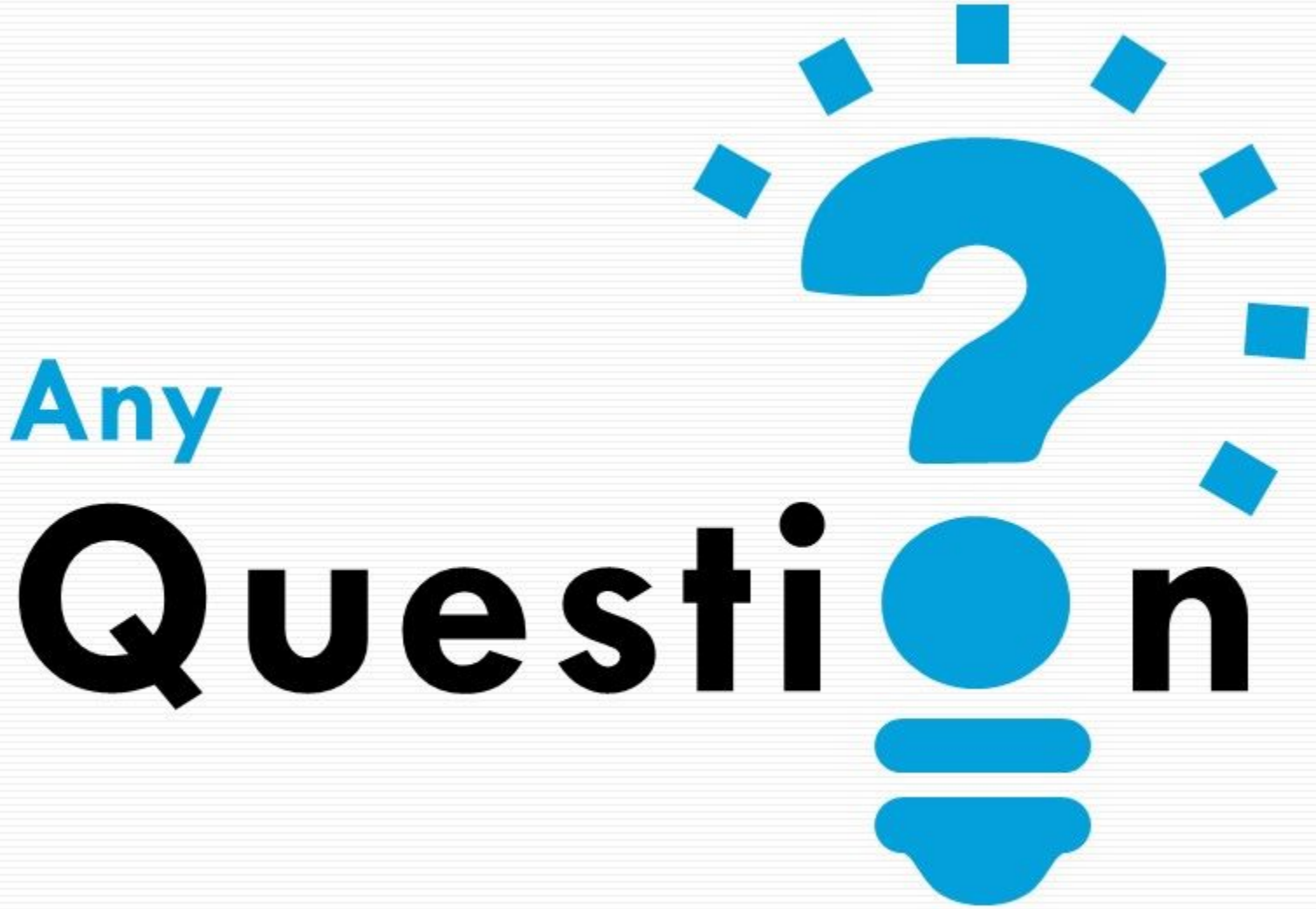
To Write Clean function

- ✓ Use descriptive names: Choose meaningful and self-explanatory names for your functions that accurately convey their purpose.
- ✓ Keep functions short and simple: Functions should ideally be kept short and focused on doing one thing well. If a function starts to become too long or complex, consider breaking it up into smaller sub-functions.
- ✓ Use consistent formatting: Use consistent formatting such as indentation, line breaks, and curly brackets to make the code more readable and organized.
- ✓ Avoid global variables: Avoid using global variables within your functions as they can lead to unexpected side effects and make debugging more difficult.
- ✓ Minimize side effects: Functions should ideally have minimal side effects and not modify any external state, as this can make the code harder to reason about and test.
- ✓ Handle errors gracefully: Always handle errors and exceptions in a consistent manner, providing clear error messages and returning early when appropriate.
- ✓ Use comments sparingly: Comments can be useful for explaining complex logic or documenting code, but overuse of comments can make code harder to read.

Tasks

- What is Callback function , recursive function and Anonymous function.
- How to convert array to string and vice versa ?

MEC ACADEMY


Any
Question

Next Session

- Strings
- Math functions
- Form handling
- File System
- JSON

MEC ACADEMY