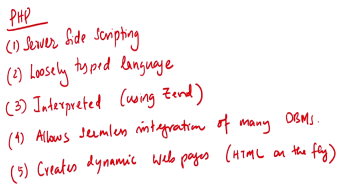
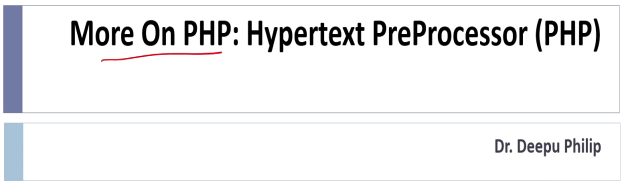
Computer Aided Decision Systems
Industrial practices using Big Analytics
Professor Deepu Philip
Department of Industrial and Management Engineering
Indian Institute of Technology, Kanpur
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology, Kanpur
Lecture 47
More On PHP: Hypertext PreProcessor (PHP)

Good afternoon, everyone. Welcome to yet another lecture on Web-based Decision Support Systems for practitioners. And, I am Deepu Phillip from IIT Kanpur. And, we have been towards the end of the last lectures of this course. And, we have covered a significant number of topics, which are related to Decision Support Systems that are intended to be used in open business problems and are trying to help the decision maker to make decisions related to unstructured open business problems.

So, if you look into this quickly as a recap, we were working on PHP as the application layer or the way we are, we are creating the MBMS (the Model-based Management System) and also the application layer that connects the Database, the User Interface, and the Model, etcetera. So, why is PHP so important? Because of its server-side script and stuff like that.

(Refer Slide Time: 01:22)





So, if you look into the slides, what we covered yesterday about PHP was that it is,

1) Server-side Scripting Language.

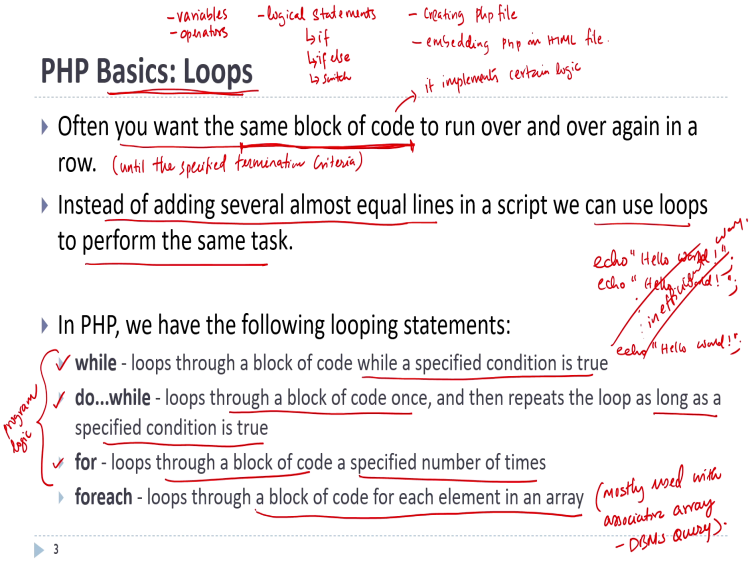
2) And, we were also told that this is a Loosely Typed Language. And, the reason is the typecasting of the variable. The type of the variable does not need to be declared specifically.
3) The third reason was, it is an Interpreted Language. And, that is using the ZEND engine.
4) And then, we also said that it allows Seamless Integration of many DBMS (Database Management System).
5) And, we also said that it Creates Dynamic Web Pages or HTML on the fly.

We mentioned these 5 points as the major reason why we chose PHP in this part. So, we are going to go more into PHP today in the lecture, and we want to see what are the additional features associated with PHP.

(Refer Slide Time: 02:52)

So, loosely on today's agenda, we will again go through some of the basics and access data types, control structures, functions, classes, and objects, and security and features. This time, we would not get into the classes or objects because we do not have time to cover that whole lecture on its own. So, we will try to stop at functions this time.

(Refer Slide Time: 03:12)



So, we were looking into many PHP basics yesterday. So, if you look into it, today, we are going to start with the loops. But so far, we have seen many of the other basics that we have seen so far include variables, we have seen operators, we have seen logical statements. These logical statements include that, if, if-else, Switch etcetera and we also saw other aspects of creating PHP file and we also saw embedding PHP in HTML file, we saw both these things. So, these were some of the stuffs in the previous lecture.

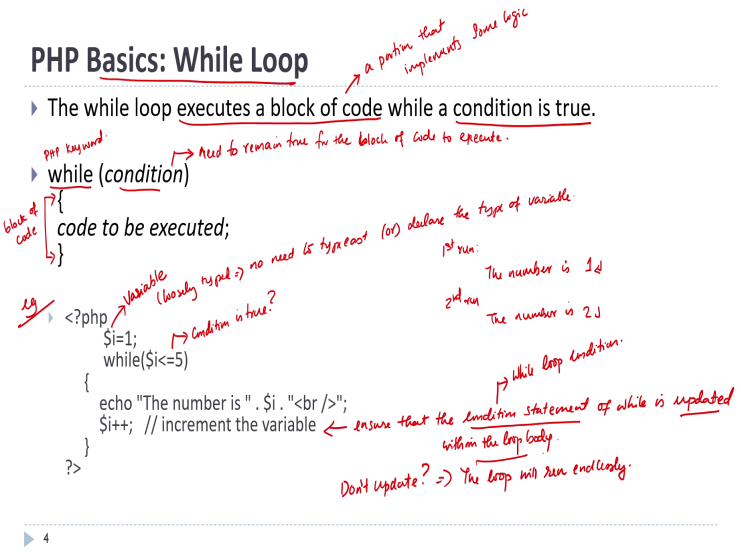
So, today what are we going to do? We are going to enhance that and we get into what we call as the loops and loops are very important in any programming language, because,

- Most of the time the programmer would like to execute the same block of code to run over and over again in a row. So, you would want a specific block of code. This block of code implements certain logic. There is some logic with this. So, this same block of code is you want to run it over and over again for some specified period of time. (So, until the specified termination criteria.) So, how do we do that?
- So, instead of adding several lines. We do not want to keep on typing the several lines, what we do is, we use loops to perform the same task. Let us say I want to print, so let us say echo "Hello World!"; let us say how to do this 10 times? One option is I do this like this and I write 10 times. You can do this, this is doable. But then, the point is, if you want to do it 20 times and you have to redo the code again or I create a scenario, so, this is an inefficient way.

The efficient way is to embed this in a loop so that it can run for multiple times. So, in PHP, we have three major loop statements and there is one which is specifically meant for associative arrays.

- So, the major loop statements are:
- While, that is, it loops through a block of code, while a specified condition remains true. So, it keeps on running until the condition is true.
- There is a do….while, which is different from a while which again remains there as long as the condition is true, the loop will run. But, the difference between While and do….while is, the do….while block will execute the code at least once, whereas the While Loop has no guarantee that at least once the code will be executed.
- And then, there is a For Loop. This is used when how many times the number of loops to be run. So, that block of code and we know it needs to be run 20 times then, put in the For Loop that is the easiest thing to do.
- Foreach Loop specifically will run through a block of code for each element in an array mostly used with an associative array. we saw what is an associative array, DBMS query, that is what you should understand specifically for, foreach and we will see all these four loop structures in a second.

(Refer Slide Time: 07:38)



So, let us start with the While Loop, the basics, While Loop.

- So, as I said earlier, the While Loop executes a block of code whatever is a block of code again. A block of code is a portion that implements some logic. So, that block of

code is executed and how long it will be executed. While the condition is true. So, the while statement, so, this is the keyword, PHP keyword, the while statement, and associated with that there is a condition whatever be the condition. And, this entire code, this is the block of code, this block of code will continue to be executed until this condition remains true. This condition needs to remain true for the block of code to execute.

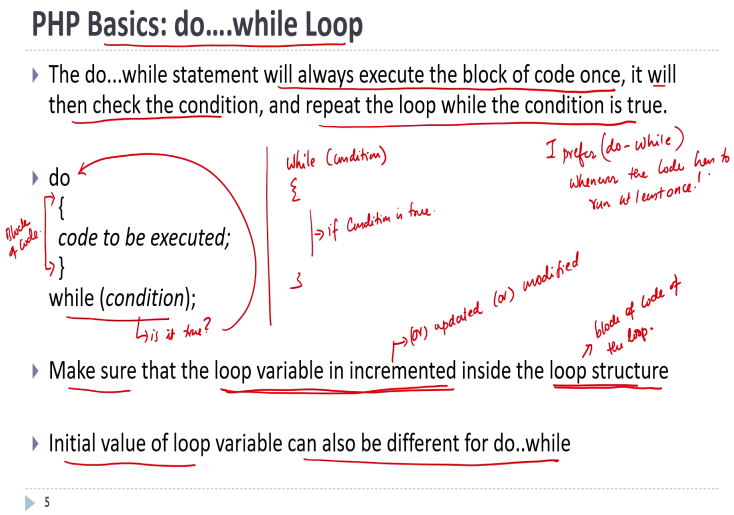
So, there are 2-3 things that you need to remember as part of this. So, let us look at an example. Here is an example of this, PHP, so we have a $i = 1; So, remember, this is a variable and this is initialized. And, again remember, loosely typed implying no need to typecast or declare the type of the variable. So, PHP immediately recognizes that this $i = 1. It takes 1 as an integer value, so, it declares $i by initial for say integer value. So then, you say while ($i < = 5), so, it checks whether it is less than or equal to 5. Yes, the condition is true. If that is true, then, print the value.

So then, what does echo do? The number is $i. So, it will first print. So, it will say the first run will print the number is $i, it will be 1. And then, there is a line break. So, it will come to the next line, second run, it will say the number is, where the $i++; so ensure that the condition statement of while is updated within the loop body.

So, you have to make sure that whatever be the loop condition that this point is the it is a variable that is incremented every time and so, the second one it will become 2, then, it will become 3 like this 4, 5 and as soon as it will print up to 5 and that the last the fifth round will be printed 5. And then, at the sixth this condition will remain false and we will exit out of the loop, but make sure that you increment or you update the While Loop condition. Which condition? The While Loop condition that used to be a condition statement needs to be updated within the body of the loop. If you do not do it, what will happen? If you do not update, what will happen? The loop will run endlessly.

If you do not want to get into an infinite loop, you make sure that the variable condition is updated within the body of the loop and then, we complete the exit out of the PHP interpreter.

(Refer Slide Time: 12:06)

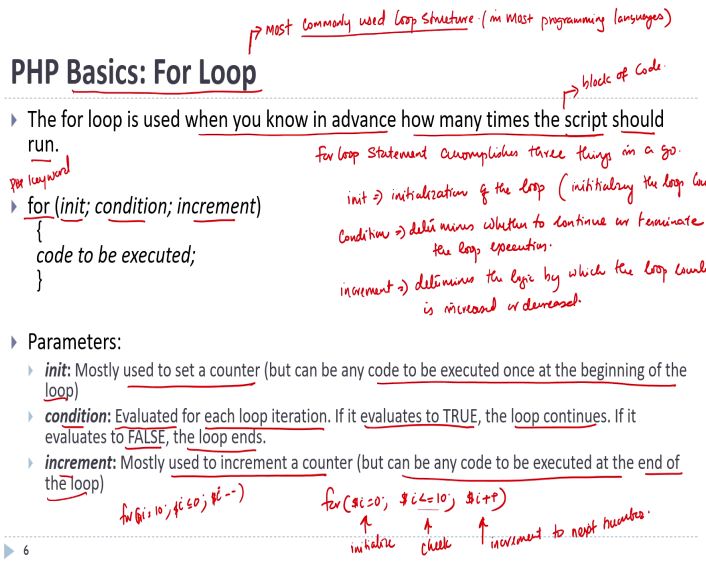


Now, let us see how it differs from what we call the do....while Loop. So, what we see is the While Loop at this point. The starting condition is while, the block of code.

- The major difference between the While Loop is that it will always execute the block of code once, whatever be the condition the block of code will be executed once. After executing the code, it will check the condition and it will repeat the loop while the condition is true. So, if you take the logic is, in contrast to the while, the while's statement is here. while, the condition and then, you have your block of code. So, the while will only execute this block of code, if the condition is true.
- On the other hand, it will execute the block of code once and it checks the condition then, after that it checks the condition, so, checking the condition is the second step. And, if it is true, is it true, if it is true, it will go back and execute the code and they will continue to do this until the condition is false.
- To make sure that you have to ensure that the loop variable is incremented or updated or modified inside the loop structure. Inside the loop structure means inside the block of code of the loop. So, you make sure that the loop variable is incremented or updated or modified within the loop structure or the block of code.
- And, the other thing is that the initial value of the loop variable can also be different for do….while. So, you can have different initial values for the do while compared to While Loop. Because, While Loop, you have to make sure that the condition needs to be true to execute the code once, whereas do, it will execute no matter what wants. So, that is the do.... while Loop. So, personally which one do I prefer? I prefer

do….while. Whenever the code has to run at least once. If the block of code requires it to be done at least once, irrespective of whatever the value is, then, I will make sure that I use the do while. If that condition is not there only then, I go to While. So, I usually prefer to do while, compared to a while.

(Refer Slide Time: 15:00)



Now, let us look at the next basics of the loop which is called the For Loop. And, For Loop is this the most commonly used loop structure. So, in any programming language, whether it is, in most programming languages, not just PHP, C or whatever it is. So, this is the most commonly used loop structure. So, people use it very frequently and why do you use it frequently?

- When you know in advance, how many times a script should be run or how many times the block of code, script a block of code, if you know how many times the block of code or the script needs to be run, then, you can use the For Loop. And, For Loop is a very interesting statement compared to the other one because the for statement, this is the keyword of the PHP keyword.
- It has three things within it. For Loop statement accomplishes three things in a go. So, it has three parameters.

- The init (initialization) parameter. So, the init implies initialization of the loop. Most of the time we also call it us initializing the loop counter. So, you mostly use this statement to set the counter and that is at the beginning of the loop statement or it can be any code to be executed once at the beginning of the loop. So, that is pretty much

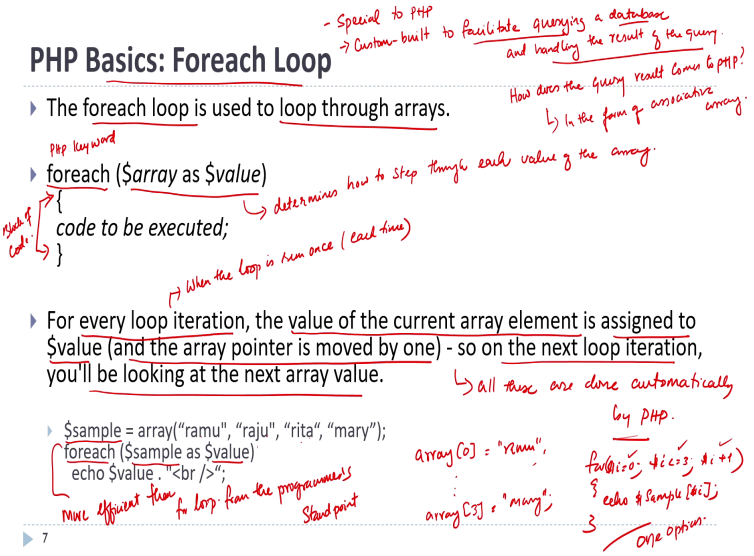
what it is. It does not have to be the initial statement; you can also put code also there, does not matter.

- The condition determines whether to continue or terminate the loop execution. So, we determine whether to continue or stop the loop execution. So, each loop iteration whenever you go through each loop iteration, this condition gets evaluated and if it evaluates to true, if the condition remains true, the loop continues. If it evaluates to false the loop ends. So, that condition determines whether to terminate or continue the execution of the loop.
- Increment determines the logic by which the loop counter is increased or decreased, can increase or decrease, does not matter how it is, does not have to be incremented all the time. It can be a positive increment or negative increment. So, mostly used to increment the counter, but you can also instead of incrementing you can also execute the code at the end of the loop that is also doable for you.

So, I can write an example: if I say for ($i = 0; $i< = 10; $i++) if I say something like this, what it says is initialization. Initialize I = 0, check. So, this is initialized and then, you do check. Check whether it is less than or equal to 10. If it is, execute the code and then, once the code execution has happened, this is incremented to the next number. You can also write another way.

Another option is for ($i = 10; $i< = 0; $i--). You can also create a loop like this, a dollar which is PHP, so we need the dollar sign. So, it starts at 10, you can decrement it also. So, both are possible with the For Loop.

(Refer Slide Time: 20:05)



Then, comes what I mentioned earlier is the Foreach Loop. And, the Foreach Loop, I told you very clearly that this is a special to PHP and custom built to facilitate querying a database and handling the result of the query. The intended use of Foreach was to facilitate the querying of the database and handling the result of the query. So, how does the result of the query come in the PHP? How does the query result come to PHP? How do you do that? In the form of an associative array. Associative array means the key value.

- So, Foreach Loop is used to loop through the arrays. That is the main aim of this one.
- Now, how does it happen? The PHP keyword is Foreach and what it does is? It says $array as $value. So, this basically says, determines how to step through each value of the array. So, what you write there, determines how you step through each value of the array. And then, this is the block of code. So, how does this logic work?
- For every loop iteration, for each one run of the loop, when the loop is run once or each time the value of the current array element, you take the current array element whatever it is, it is assigned to dollar value. So, it goes to the array, finds out whatever is the current array element, it gets assigned to the variable dollar value and then, the array pointer is moved by one. So then, the pointer is moved to the next one. So, at the next loop iteration, you will be looking at the next array value. All these things are done automatically by PHP. You do not have to keep track of that endpoint or you do not have to keep track of the index or anything of the sort, PHP will do all of this for you. This considerably simplifies your programming effort.

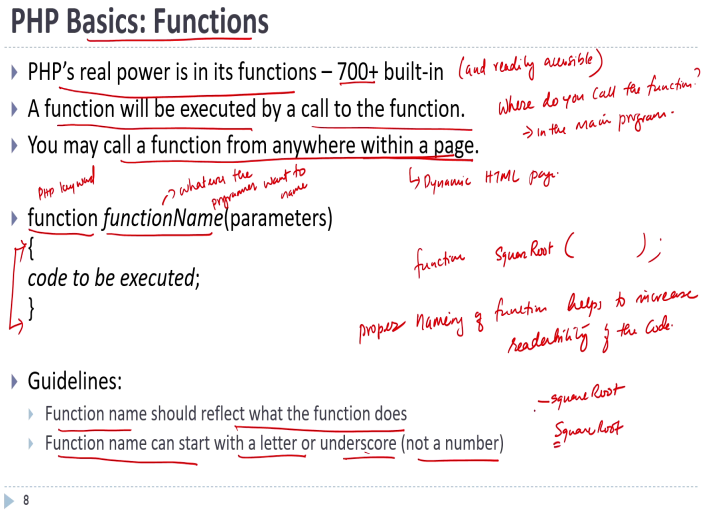
So, let us for example take a dollar sample as an array of Ramu, Raju, Rita and Mary. So, you can think about it as array [0] = “Ramu”, etcetera like this and array [1] = Raju, 2 is Rita,

array 3 is Mary, that is what happens. Now, you can write a loop. One way to do it is ($i = 0; $i < = 3; $i++) and you can say echo $sample [$i]; I can do this as one option. So, I need to decide what is the start? What is the maximum value, the increment everything I can do this? This is one option. Instead of that, I can just write one line of statement, which says Foreach dollar sample dollar sample is your array as dollar value.

So, it says Foreach value in the array dollar sample, take one, assign it to dollar value, and then, step to the next and then, only you can use equal dollar value. So, I do not need to know what is the starting counter, what is the ending counter? What are the incrementation criteria? I do not need to know any of these.

In just two lines of code, you will step through each element of the array. So, this is more efficient than For Loop, from the programmer standpoint, not writing large or huge numbers of code, huge lines of code. So, life is easy with you in this regard. So, most of the time when you do a query result you will use Foreach Loop.

(Refer Slide Time: 25:36)



Then, we get into what we call the basics, which is the Functions. What are PHP Functions?

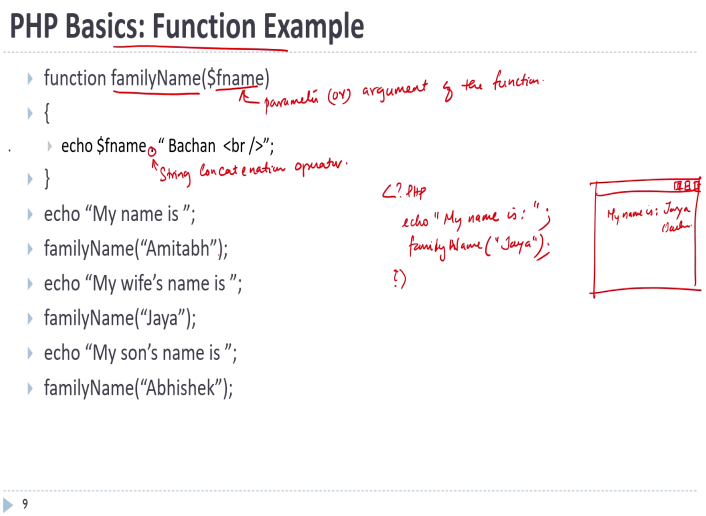
- And, one of the real powers of PHP is in its functions. Now, this number is beyond 1000. And, every day, I am pretty sure at least 10 functions get added to it, 700+ really powerful built-in functions, built-in and readily accessible. You do not need to write, if you want to do a decision tree algorithm or you want to do what you call as a random forest or K means clustering it, all these things are built-in PHP. Somebody

has written this code, and it is available readily for you. So, how can you use these functions?

- So, the function will be executed when you call the function. So, where do you call the function? Where do you call the function?
- In the main program, where on the part of the main program we call the function or main program or you may call the function anywhere within a page or whenever within a dynamic way. So, which page? This is the dynamic HTML page. That is the page that we are talking about at this point.
- So, how do you declare a function, you use the keyword function, so, this is a PHP keyword. And, the function Name, so, typically, this is whatever you want to name, the programmer wants to name. If you are writing a function ending the square root, so, you may say a function square root ( ); and put the parameters and that would be a function. So, you will say function square root, and then, within that function, within two curly braces, you put whatever be the block of code to be executed.
- So, the couple of guidelines is that the function name should reflect what the function does.

- You should not name a function that finds the square root of a number as My Function or is like a Weird Function. So, those name in such a way that, look really the name of the function, what it is, and this name of the function also, when you call it in the program, the proper naming of function helps to increase the readability of the code, that is a very important criteria from your side. So, when you write the code, and when you call the function, you see the code name and then, or the name of the function and you understand what that is.
- And, just like a variable, a function can start with a letter or an underscore, you cannot start with a number. So, as I said, you can write a function name that can start with an underscore square root or something like this. This is doable. Or you can start with a capital letter or capital, small does not matter, or an alphabet can be either with an underscore or a letter, and you cannot start with that number.

(Refer Slide Time: 29:04)



So, here is an example. So, here the key word is Function example. The key word is familyName. So, it is some family name. So, whatever you give the first name of the individual a family name, a specific family name gets added to it, the variable ($fname) is called the parameter or argument of the function. The parameter or the argument of the function, so, what does it do? It takes the f name, first name and this is the string concatenation.

Remember, we studied this yesterday, the string concatenation operation or operator. So, it adds Bachan, Amitabh Bachan's last name to this easy way to showcase. So, in the main program if I call the family name Amitabh, and then, what it does is, it will print Amitabh Bachchan, add the family name Abhishek, it will add, Bachan with it and says Abhishek Bachchan, that is what the program will output.

So, if I say something like this:

<?PHP

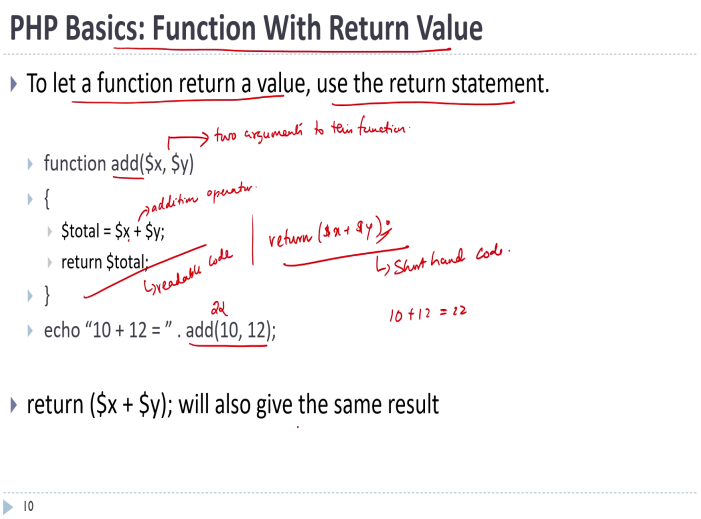
Echo "My name is:";

familyName ("Jaya");

?>

So, what does it do? It will print, it will go there and print Jaya Bachan. So, the output will look like on the screen, the browser will show like this. It will show that my name is Jaya Bachan, whatever. So, you got the point. So, in this case, the function does not return anything, the function just prints it and calls it good.

(Refer Slide Time: 31:14)



So, the functions can also return a particular value.

- So, at the end of the execution of the function, if you want to let the function return a value to the main program, then, use the return statement.
- So, we declare a function called add and here are two arguments to this function. The first is $x and $y. So, what does this code do? It takes $total = $x + $y; So, this is the addition operator. So, it takes x and y, sum the values, put it in total and it says return $total; So, return means the function will send this value back from wherever it is called. So, if you say echo "10 + 12 =" and then, this is the string concatenation operator add 10, 12, you call the function, so, the total gets returned here. So, this function will return the value of 22. So, it will show us 10 + 12 = 22 this is what it will be shown.
- You can also, instead of doing all of this, this can also be replaced as return dollar x plus dollar y, this also is fine, you can change it into a single line of code and get it done. And, that will also give you the same result. But this is more, this is shorthand, and, this is readable code, this shorthand code, both of them will end up giving you the same result.

So, with this to the conclusion of the advanced portions of PHP.We stop here purely because of the fact that they should give you sufficient enough stuff to learn and start working on a simple application as part of the Web-based Decision Support System. Now, in the coming lectures, we will see how we can access or we can use PHP to access the MySQL database and also, we will see how to access the HTML web page. Thank you.