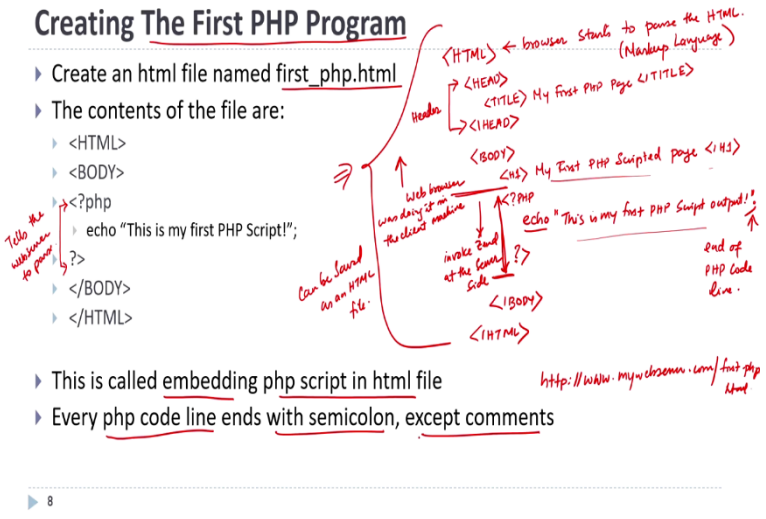
Computer Aided Decision Systems - Industrial practices using Big Analytics

Professor Deepu Philip
Department of Industrial & Management Engineering
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology, Kanpur

Lecture 46

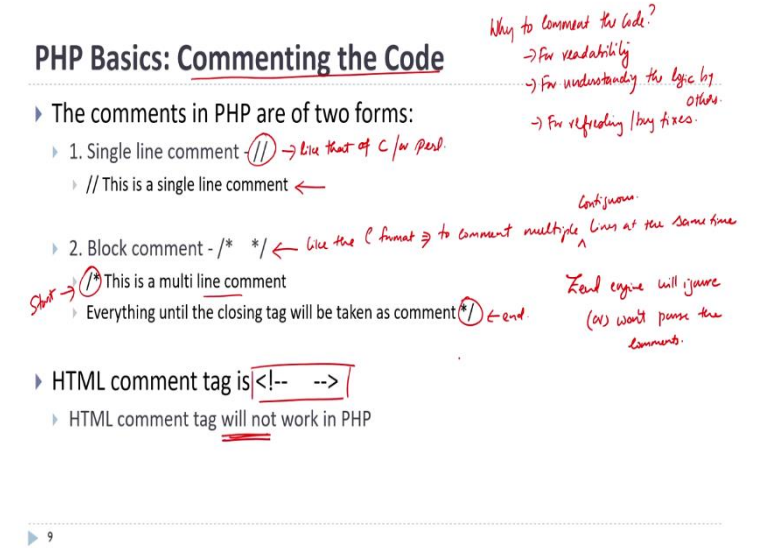
Basic Components of PHP

(Refer Slide Time: 00:14)



Continuing where we stopped previously, looking into The First PHP Program, and we want to talk about the Comments. How do we Comment the PHP code?

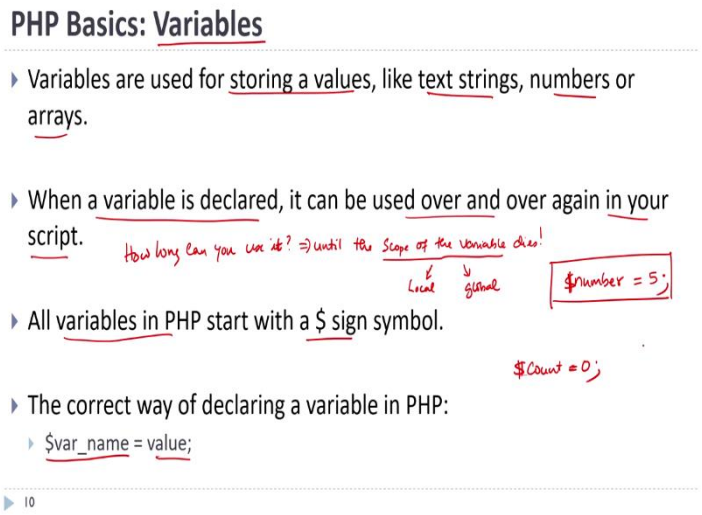
(Refer Slide Time: 00:19)



So, the basics, how do you comment the code? So, PHP comments, the program comments. Why to comment the code? That is for readability and for understanding the logic, by others. It can also, for refreshing or bug fixes. We want to update, refresh, etcetera, all those kinds of things, you end up commenting the code. So, committing the code is a very good practice and it should be followed. And, the two ways you can do it,

1) Two continuous slashes (//) that of C/Perl, the same approach, two continuous slashes.
- This is a single line comment, and can only comment one line.
2) If you want to comment multiple lines, this is also like the C format to comment multiple lines at the same time (/* & */). So, this multi-line (/*). So, everything that is within the start and end. So, this /* the start and */ is the end. So, everything within this will be taken as the command. And so, the ZEND engine will ignore or will not parse the comments. It will ignore or it will not parse the comments.
3) Remember one thing, the HTML comment is in this format (<!-- -->).
- It will not, this command approach will not work in PHP. So, if you want to do an HTML command or something like that, create that, then, that has to be within either this format, the two parallel lines, two parallel slashes or this slash asterisk format. The HTML tag is not applicable in this.

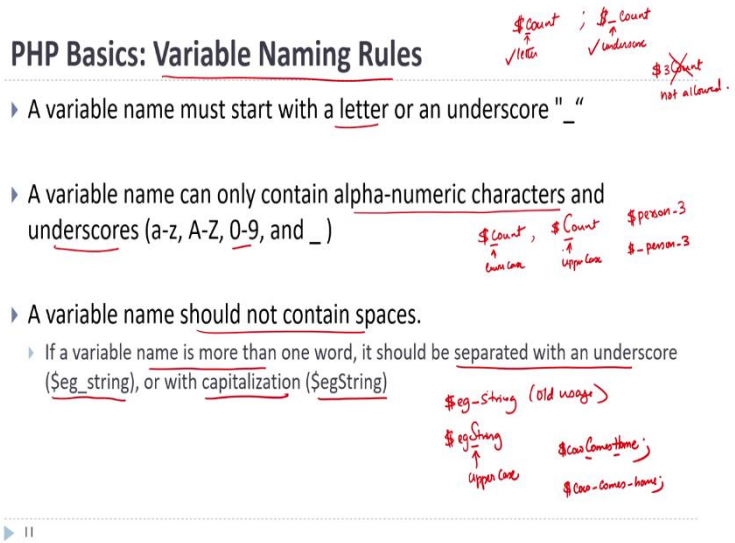
(Refer Slide Time: 02:56)



Next, the major basics are variables. What PHP variables. So, variables are, they can take any value.

- Variables are used for storing values. It can be text strings, it can be numbers, it can be arrays, it can be float values, etcetera. So, the main name of a variable is to store value, something.
- In a PHP script, when you declare a variable, it can be used over and over again in your script. Once you declare a variable, you can use it until the scope of the variable dies away. So, how long can you use it? Until, the scope of the variable dies. So, we will talk about what is the scope of the variable a little later. So, there are typically two ways. One is local and the other one is called global. You might have studied these local variables and global variables, that is what we are called as a scope. Local variables are typically to a function, it is local to a function. So, once you exit out of the function, the scope of the variable goes away. Global is for the entire program. So, until the program execution is terminated, that variable remains in the memory.
- The critical thing is that all variables in PHP start with the $ sign. So, if I want to create a number, a variable, typically in C you create a variable called number and I say it as number equal to 5, that is doable in C. In PHP, there is a $ before this. So, it will be a $. So, this $ will be the integer variable I did in this regard. So, the current way of declaring a variable in PHP is ($ variable name = value;). So, if I say $ count = 0; if I say that count is a variable, PHP variable and it is initialized to the value of 0. That is all a variable gets declared in PHP.

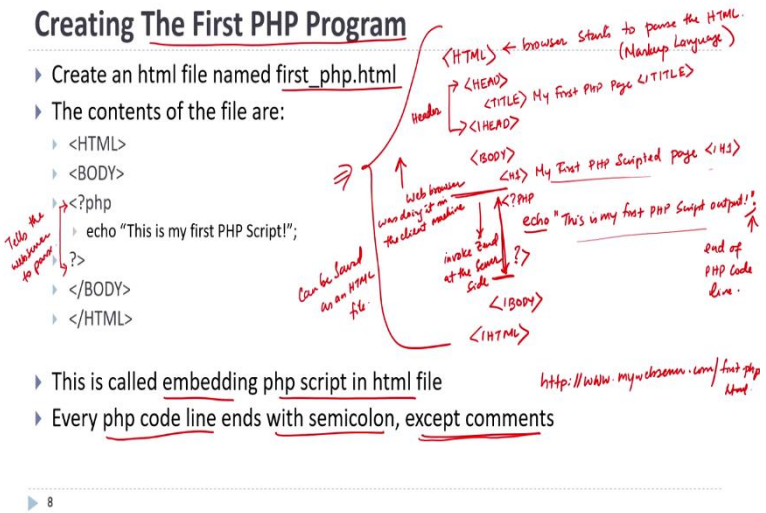
(Refer Slide Time: 05:01)

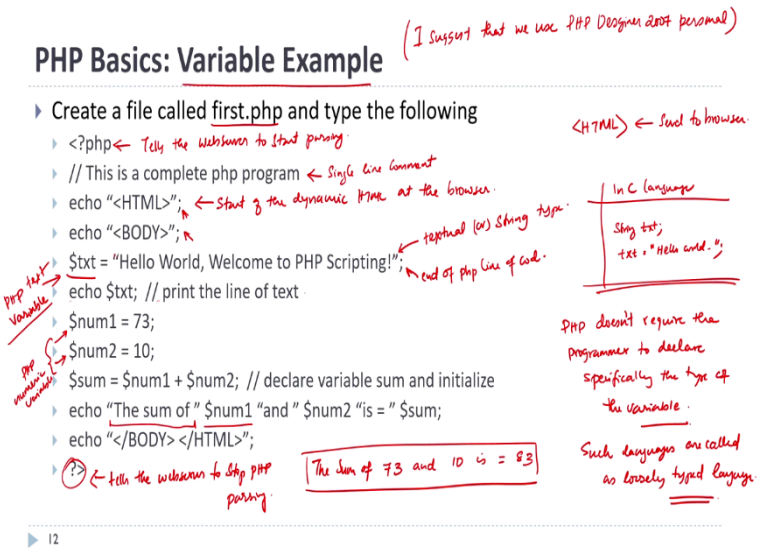


Now, what are the variable naming rules? There are some things that you cannot keep on writing, whatever nonsense you want to write. So, there are a few things that you can say.

- The variable name must start with a letter or an underscore. So, I can write like this $ count. This is a letter. I can also write it as a $_count. This is also possible; it is an underscore. Both of these are allowed. But, if I do this $ 3 count, not allowed. I can only start with either a letter or an underscore.
- And then, a variable name can only contain alpha-numeric characters and underscores. And, not in the first position, but you can use numerals also. So, I can create a variable like this, $ count or I can create another variable $ Count. One is lowercase, and another is upper case. Both are okay. I created a variable called $ person_3. That is doable. $_person_3, that is possible. So, these both are okay. The numeric values 0 to 9 can only come after the first character, the starting character, letter on underscore.
- Then, the third is, a variable name should not contain spaces, cannot put spaces in between. That is a no, no when it comes to variable names. So, if your variable name is more than one word, it should be separated with an underscore. So, like an example $eg_string or with capitalization $egString. So, I usually follow the second format because the underscore is difficult. Sometimes, you miss the Shift key and it becomes a problem. So, you can write something like this $eg_string. This is what old usage mostly was. A lot of the people used this. The new usage is $egString. This is uppercase. If I create a variable $Cow _Comes_Home; then I will do it as $CowComesHome. This is also doable. Both of these things are acceptable, but having this underscore is slightly difficult. So, a lot of the time, the new ones we separated with the help of capitalization.

(Refer Slide Time: 08:12)





So, here are some examples. Example code of the variable. So, now what you are going to do is, you are creating a file called first.PHP. The previous example that we have gone through, it is called the first_PHP_HTML. We created it as an HTML page. Now, we are creating a PHP page and to do this I suggest that we use PHP Designer 2007 Personal. It is a free software ID and then, type the code in this. So, the first line in this is PHP. So, this means it tells the Web Server to start parsing. So, then, this is a single line comment. It is just a comment. This is like a complete PHP program for somebody to read and understand.

Then, what you wrote is echo HTML. So, when the PHP is creating a dynamic HTML, first it will print this HTML, this will be sent to the browser. Now, the browser will say there is an HTML, that means I have to show HTML, I should be in the way of doing a markup language.

Then, comes the next one, echo BODY. So, HTML is the start of the dynamic HTML at the browser. The browser we begin and then, the next one says BODY. So, there is no header, it is just BODY.

And then, you have a variable, $txt, this is a PHP variable. And, what you are saying is, in that the variable is within double inverted commas, it says "Hello World, Welcome to PHP Scripting!". So, anything inverted in this says, it says textual or string type.

So, one thing that you should probably notice by this time is, in C language, we used to do this,

String txt;

txt = “Hello World.”;

PHP does not require the programmer to declare specifically the type of the variable. We do not have to say whether it is an integer, string, float, etcetera. None of this is required. So, whereas in C, it is required. You declare what is the type of the variable. So, such languages are called loosely typed languages. So, the variable type casting is not a very strict thing in PHP. So, when the PHP says "Hello World, Welcome to PHP Scripting" within the two inverted commas, it will know this guy is putting a string.

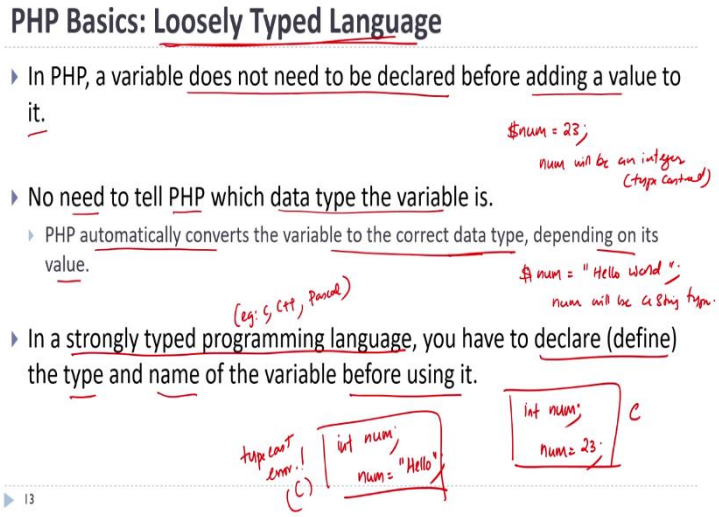
So, it will be declared by itself as a string. And, this is the end of PHP line, end of PHP line of code. And, this is true for the end of the PHP line. The semicolon marks the end of the line of the PHP code.

Then, we say echo txt and we have a comment here. It says print the line of text. So, it will print. "Hello World, Welcome to PHP Scripting". So, this is a PHP variable and this is a PHP text variable and you have $num 1 and num 2. These are PHP numeric variables. That is what this is. So, 73 and 10 are the two numeric variables that we declared. And then, we say the $sum = $num 1 + $num 2. So, what it does is, just take 73 and 10, it will add and store the value of 83 in the $sum and then, it will print echo the sum of within this. So, first it will print this string, then, num, So, it will print as the sum of 73. The $num 1 will be 73, and then, print and 10, $num 2 will be 10 is equal to sum will be 83. So, this whole thing PHP will create for you from this.

And then, you say echo slash body slash HTML, that means it is closing the HTML stuff. And then, it says this is the end, that tells the Web Server to stop PHP parsing. So, that is the

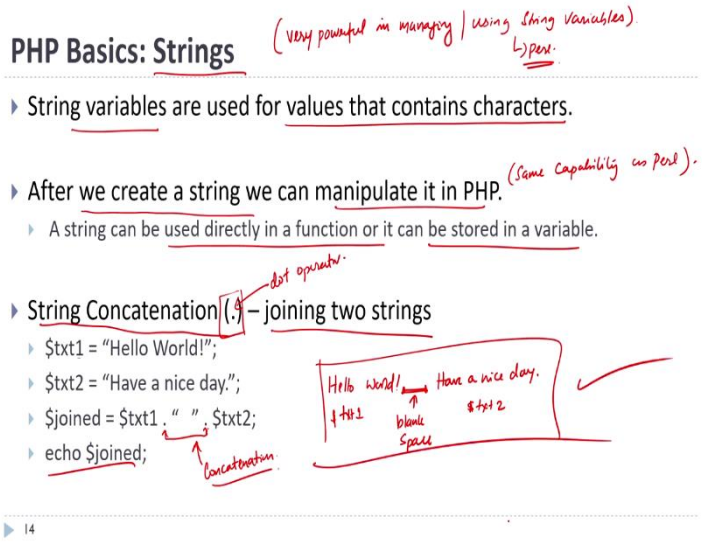
important aspect of this PHP code. And, we have now seen how both text and the numeric variable are used in this script. So, as I said earlier, it is a loosely typed language.

(Refer Slide Time: 14:12)



- PHP is a loosely typed language because in PHP a variable does not need to be declared. You do not have to declare the variable before adding a value to it. So, you can just declare a variable and add a value and PHP will decide what type it is and add the type casting to it.
- So, there is no need for you to tell PHP which datatype the variable is.
- So, what happens, the PHP automatically converts the variable to the correct data type depending on its value. So, whatever the value you assign. So, if you say $num = 23; then, num will be an integer. It will be type casted to an integer. If I say, $num = "Hello World"; then, num will be a string type. So, the PHP looking at the value of the data type, it declares based on this value determines the data type of the variable. So, that is why PHP is called a loosely typed language.
- In a strongly typed programming language like example: C, C ++, then, you can think about Pascal, all these things. They are all strongly typed programming languages. You have to declare or define the type and the name of the variable before using it. So, if it is in C, you will have to say int num; and then, you say num = 23. You can do this, this is C. If I do this int num; and I say num = "Hello", C will throw an error. This will give a typecast error. You have used the wrong typecast of the variable. So, this will be given by C in this regard. So, PHP you do not have to worry about because the language in itself will do whatever is the appropriate type casting.

(Refer Slide Time: 16:29)

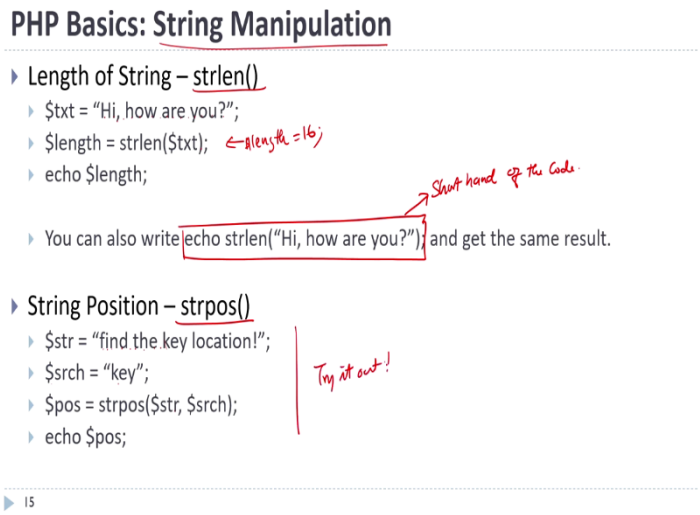


Now, let us talk about the PHP strings and the PHP is very powerful in managing or using string variables. PHP has a lot of power and this generates it from Perl. Perl also has the same string capability power. So, what is a string?

- String variables are used for values that contain characters. When you have a large set of characters, you use string variables.
- So, after we create a string, we can manipulate it in PHP, the same, as that of the same capability as Perl.
- So, a string can be used directly in a function or can be stored in a variable.
- So, one of the important things is that an operation is called a string concatenation. Concatenation means joining two strings and that is done with the help of this dot operator (.). So, if you look at an example, $txt1 is "Hello World" text 2 is "Have a nice day".

Now, what I am doing here is $joined = $txt1. " ". $txt2; So, what does it do? It does "Hello World" and to which it will add a space here. Then, it will add, "Have a nice day", "Hello World!" , "Have a nice day". So, these two variables, this is your $txt1, this is your $txt2. These two variables are concatenated with a blank space. So, (." ".) are your two concatenations. Concatenation or joining of it, right? And, what you do, when you say eco joined, So, this whole thing is printed, that is what the one capability of PHP.

(Refer Slide Time: 18:41)

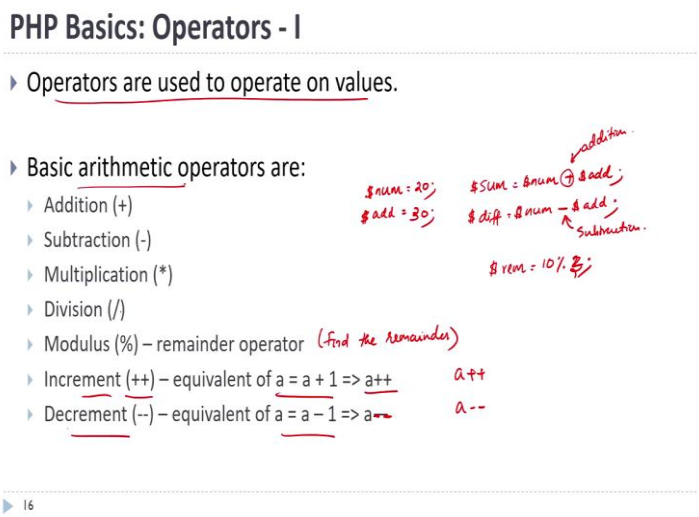


Now, we can also do other string manipulation and as I said, PHP has a lot of string capabilities like Perl.

- So, find the length of the string. Strlen() string length is a function that is embedded in PHP. So, we have a variable called $txt = "Hi, how are you?"; And then, $length = strlen($txt); So, what does it do? It takes this text and says 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, it even counts the blank space also. And, it says I counted 16. And, that value of length gets stored at 16. So, length will be equal to 16 and then, equal length. So, it will print the value of 16. It says that the length of the string is there, or you can also write echo strlen. This whole thing. Instead of doing it, you can combine it into a single line. This shorthand code, short hand of the code. So, you can reduce three lines and combine everything into one go and you get the same result.
- So, another example is, there is another thing called, another example. I am calling it a strpos. A string position. So, what the strpos or string position says is that it can find where particular string is located. So, the $str is a string variable. Find the key location and search variable is the key.

So, I am saying that $pos = strops(str, $srch). So, what it does is, it will do whether this key, where is, which position it is. So, it will say here is a string and will search here is key. So, 1, 2, 3, 4, 5, 6, 7, 8, 9. So, it says from the tenth position onwards the key, string, is there. So, you can say that the count of tenth, the eleventh position, you can see the keyword key is available. So, this is an example code. Try it out by yourself and see whether it actually works.

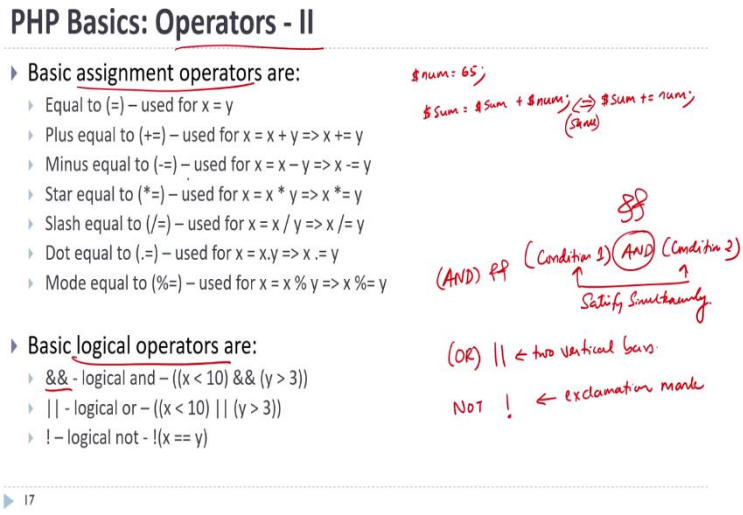
(Refer Slide Time: 20:45)



So, now comes the basic operators of PHP, the main mathematical operators.

- So, operators, what do they do? They are used to operate on values. So, the basic Arithmetic operators in PHP are: if you have multiple values, So, addition, So, if you say $num = 20; $add = 30; So, if I say $sum = $num + $add; So, this + is the addition operator, this is the addition operator. If I say $diff = $num - $add; then, that is a subtraction operator.
- Same way, (*) is for the multiplication,(/) is for the division, (%) or it is called as the modulus- remainder operation. So, this basically says find the remainder. So, if you say the $rem = 10 % 2, if you do this, then, what is the remainder of 10 divided by 2? So, there is no remainder. If I do 10 % 3, then, the remainder value will be shown to us 1.
- Then, increment operator (++). That basically is equivalent to saying A is equal to a plus 1. Instead of that, if you say a plus plus, which means it is equal to saying that, A is equal to a plus 1. So, whenever you have to add 1 or subtract 1, then, you can use a++or a--. This looks like a large dash. These are two minuses. So, this is two dashes like this, minus, minus, that is a decrement operator. But remember, increment and decrement will only add one value to it, 1, just individual value.

(Refer time slide: 22:52)



Second set of operators, the Assignment operators.

So, this (=) basically check whether x is equal to y or something. Or, if we use this Assignment operator. So, like a $num = 65; So, that is the equal to operator. So, take the value of 65 and assign it into the variable dot or num.

The (+=) to is the short form of, so, if I say 4sum = $sum + $num; let us say if I have this, I can always say instead of this. I can say it as $sum += num; I can write this which is exactly the same, it will give you the same result. So, that is a plus equal to, that is sum is the variable, add some value with the same variable.

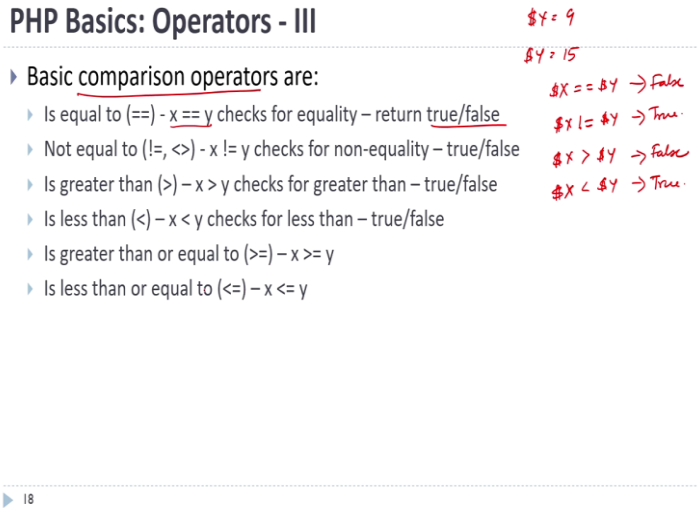
(-=) to is exactly the same, but instead of the sum it will be subtraction.

(*=) is for the product, (/=) for the division, (.=) is equal to x = x.y. The concatenation that also is doable. And then, (%=) this is also applicable to the PHP. So, these were the Assignment operators.

Now, comes the Logical operators. So, if you want to say AND, which means condition 1 AND condition 2, we have two conditions to be satisfied simultaneously. Then, this AND is represented by PHP with two ampersand signs (&&). That is a logical AND. That means both the closest are correct. The OR is given by two vertical bars OR, AND is to this one, OR is two vertical bars (||). The vertical bars are on the keyboard, right above the enter, next to the curly brackets. And, the NOT that is denoted by exclamation mark (!). Exclamation mark is a logical

NOT, NOT x equal to y. So, you are basically saying that if x and y are not equal then, that is, what it is.

(Refer Slide Time: 25:27)



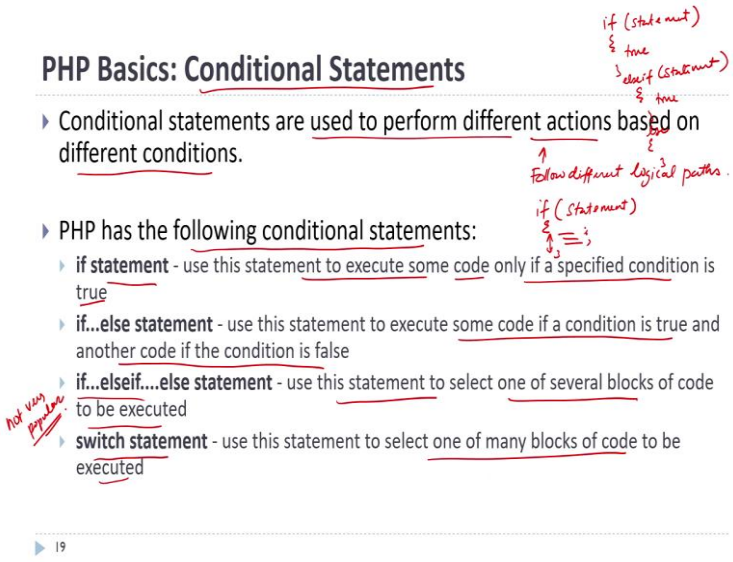
Now, the third one is Comparison operators. So, we have seen that Arithmetic operators is one thing, we have seen. Then, we have Assignment operators, we have Logical operators and the third set of operators is Comparison operators.

So, if you say x == y, if you do this, these checks, it gives you true or false. So, if we say $x = 9; $y = 15; If you say $x == $y, the answer here will be false because x and y are not exactly the same.

But, if I say $x ≠ $y, the answer here will be false. Here, the answer will be true because x and y are not equal, same thing.

Then, if I say $x > $y in this case, which is 9. Is 9 greater than 15? The answer is no, it is not true. So, it is false. If I do this, $x < $y, it will be true. So, that is the other aspect of this one. Then, the greater than or less than two is another condition. So, if you say that in a loop condition or something, we use it to check, is it value less than or greater than? That is useful.

(Refer Slide Time: 26:48)



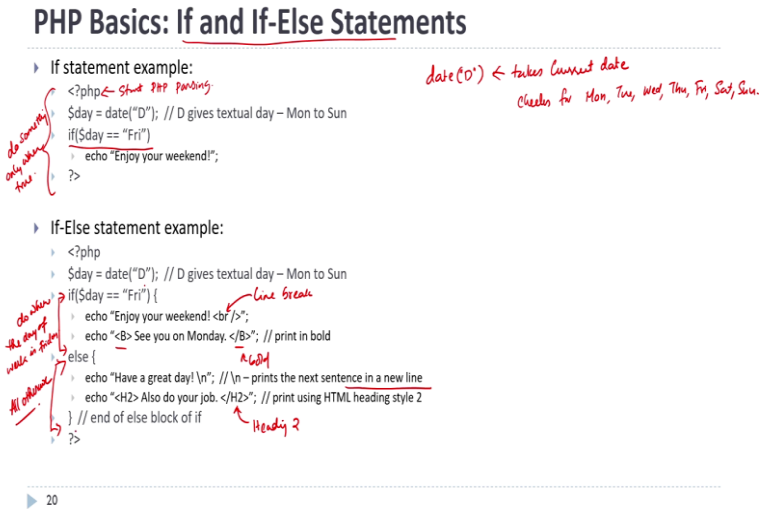
So then, another aspect is PHP has a lot of conditional statements.

- Conditional statements are used to perform different actions based on different conditions. So, you want the code to do or the program to follow different logic. These actions follow different logical paths, based on certain conditions.
- So, the following condition statements exist in PHP.

- The first is called the if Statement. So, if you write someplace if (Statement) and then, you give a set of stuff here. So, then, if this statement is true, only then, these conditions will be executed, otherwise it will not execute.
- So then, another one is the if..else Statement. So, if Statement is only used to execute some code or if a specific condition is true, that is it. Condition is true, do this, otherwise do not do anything. If..else Statement is particularly used in some code if a condition is true and executes another code if the condition is false. So, there is a scenario where something is true, then, you have to do logic. If it is not true, then, you have to do some other logic.
- Then, the third one is if..elseif. So, you can think about it as an if Statement, if this is true, do this, else-if Statement, then, it is true, do this, Else, do this. So, you can have a large set of statements based on this use of else-if usages. So, use this statement to select one of the several blocks of code to be executed. So, based on multiple conditions, that is feasible, that you can do. But, this is not very popular among programmers. People do not use it too much.
- Now, the last statement is a Switch Statement. So, a Switch Statement is, when you have to select one of the many blocks of code to be executed. So, instead of Else-If, a

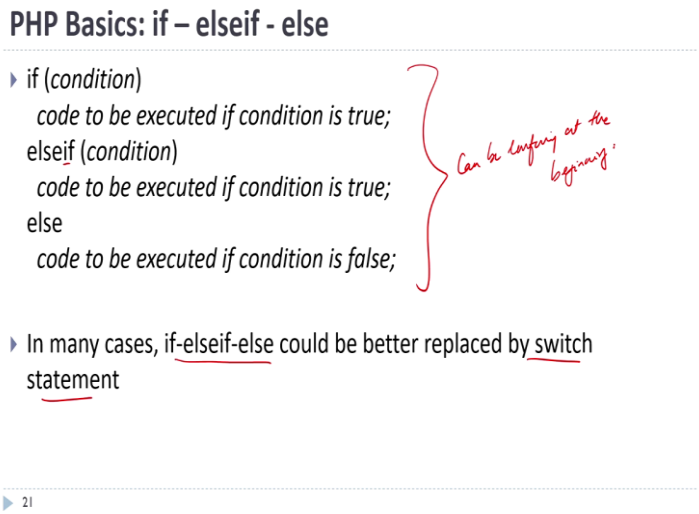
lot of the people actually prefer Switch Statements. So, I will show you an example of each one of these in the following slides. So, that would actually make your life much easier.

(Refer Slide Time: 29:12)



- So, let us look at the If and Else-If Statement. How do we code that in PHP? So, the simple example of PHP here is, the start PHP parsing. This will generally start parsing it. So, there is a function in PHP called date (*D*). So, the date function of PHP, if you said date 'D'. So, what it does is, it basically takes the current date, then, it checks for Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, or Sunday, the three-letter abbreviation and then, whatever it is, and then, returns the current day. So then, we say if $day == “Fri”, if the value that returns is Friday, then, you should print enjoy your weekend, otherwise do nothing. So, this is the, do something only when true. So, print enjoy your weekend only when it is a Friday, otherwise do not do anything.
- And, if..else example is this. Again, the same date function. If $day = “Fri”, enjoy your weekend this much. So, this is, do when the day of week is Friday, when it returns Friday, print this. Else, in all other cases, this is all otherwise, in all other cases, print have a great day, do your job. But you can also see that I can also embed tags in the HTML, HTML tag here. This is a line break. Here is bold, here is heading two, etcetera. Then, this slash 'n' is a new line. It prints the next sentence in the new line, that is the idea. So, all of this and then, the PHP parsing gets stopped here.

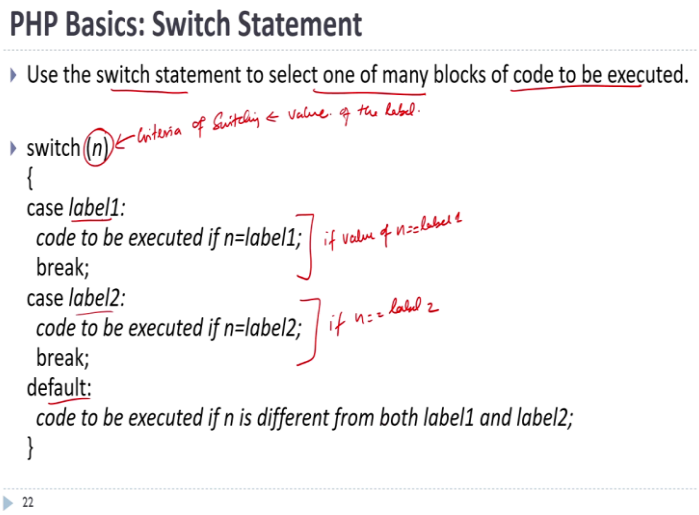
(Refer Slide Time: 31:34)



Then, If-elseif I told you, if condition is true elseif this code be executed, if the, this if condition is true else the code to be in this, if this particular is falls, etcetera, like that.

- In many cases the If-elseif-else could be better replaced by the Switch Statement. So, as I mentioned, this is not a very popular approach to do it. This can be confusing at the beginning. So, hence, it is probably better for you to use the Switch Statement.

(Refer Slide Time: 32:12)

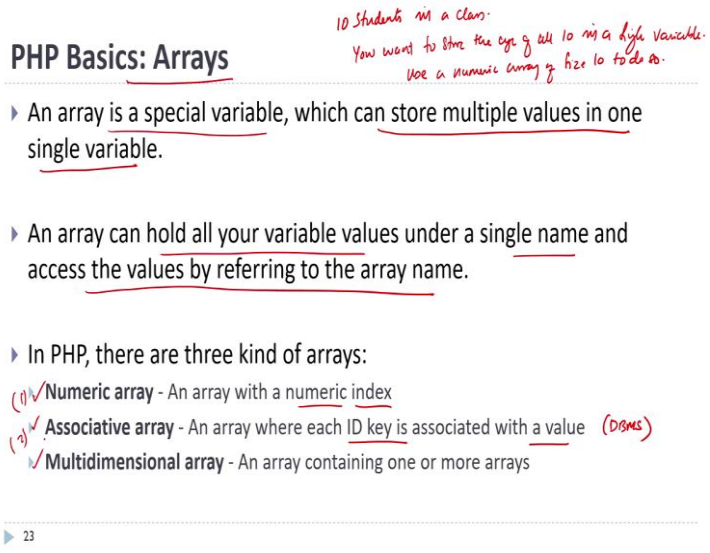


So, what is the Switch Statement?

- So, the Switch Statement is, you use it when you are to select one of the many blocks of code to be executed, when you have multiple blocks of code to be executed and you have to select only one.

So, you give a switch (n). This is the criteria of switching. This is the value of the label. And, if the value of the label is label 1, label 2, etcetera. So then, it will execute the value. If the value of 'n' is equal to label 1, it will execute this particular block of code. If the value of 'n' is equal to label 1, this is if 'n' is equal to label 2. If this is not true, then, execute default. There is a default clause, if none of the label cases are true, then, just do the default code. So, that is the Switch Statement.

(Refer Slide Time: 33:17)



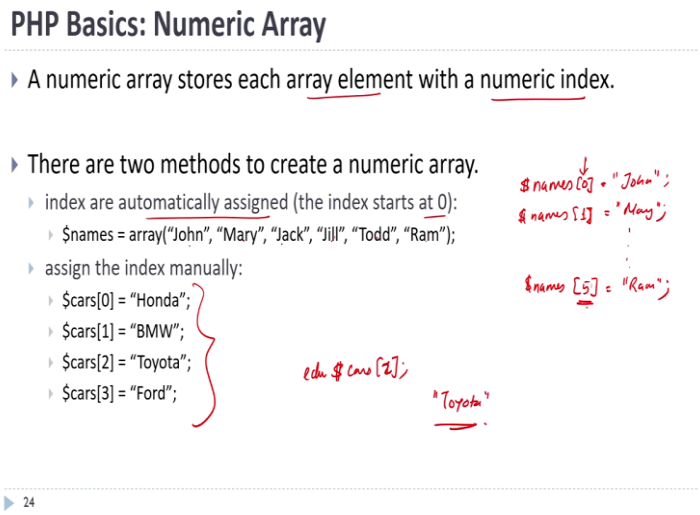
We also now, without going into too many examples, because we can see too many examples coming up, we now go into, what we call as Array, PHP array.

- So, an array is a special variable. So, what is an array? Like in all programming languages, array is also available in PHP. The advantage of the array is that it is a special variable which can store multiple values in one single variable. So, let us say you have 10 students in a class, you want to store the age of all 10 in a single variable. Use a numeric array of size 10 to do so. So, that is, what an array. In a single variable, you can store multiple values.
- So, an array can hold all your variable values under a single name and access the value by referring to the array name.
- And, in PHP there are three types of arrays:

1) Numeric array, very popular, which can hold a numeric value and actually, it has a numeric index rather than numeric value.

2) Associative array, where each ID, there is a key associated with a particular value and it is very popular with queries when you deal with the DBMS, associated arrays are very popular with this.
3) Multi-dimensional arrays are containing more than one other.

(Refer Slide Time: 34:52)

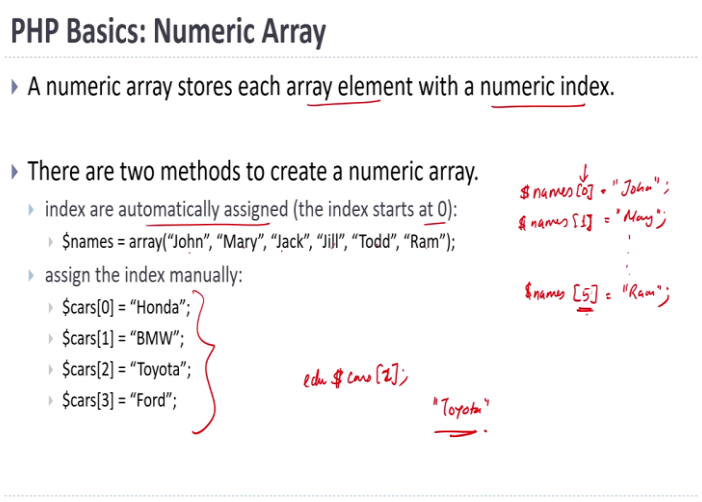


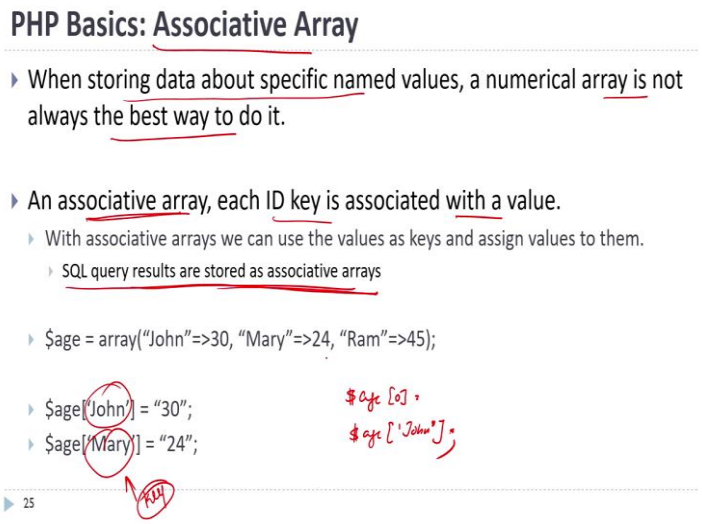
So, here is an example of a numeric array.

- A numeric array stores each array element with a numeric index. It does not have to be a numeric value, but the index is numeric
- The two methods you create a numeric array,
- if indexes are automatically assigned, you do not have to give the index, it starts at 0. So, we say dollar names equal to array, John, Mary, Jack, Jill, Toe, then, it will have $names [0] = "John"; $names [1] = "Mary"; and $names [5] = "Ram";

Remember, there is 1, 2, 3, 4, 5, 6, 6. But, this index is 5. The reason is, because the index starts at 0, the default index starts at 0. So, it will always be, the last will be stored with one value less than this. You can also assign it manually. This is another one, car 0, cars so. If I say print or echo $cars [2]; it will print Toyota. That is what will happen in this case.

(Refer Slide Time: 36:14)



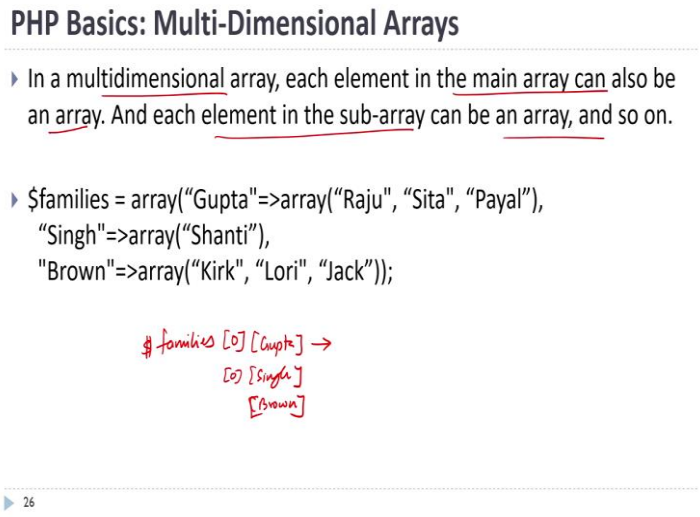


Then, comes what you call an Associative array in PHP.

- An Associate array is powerful purely because of the fact that when storing data about the specific named values, a numeric array is not always the best way to do it. Because why? You need to remember what the index is. So, if I want to know what Toyota is, I need to remember cars 2 is Toyota.
- But to solve this problem, you use an associative array. Each ID key is associated with the value. For example, if you want to store the age of the people, so then, I can basically say that $age, I say that array John 30, Mary, if you use this part, then, what I can say is that instead of declaring $age [0] = John; I can say $age ['John]; If I do this, then it will return the value of 30.

So, in such types of arrays, instead of using a numeric value as an index, you can use a key. So, this is the key. This kind of array is known as an associative array. And, it is very, very popular when SQL results because almost all SQL query results are stored as an Associative array in PHP. So, you can use an Associative array mechanism to actually access the content of the SQL query that is returned to you in PHP. And then, the last one is a multi-Dimensional array.

(Refer Slide Time: 37:45)



- In a Multi-Dimensional array, each element in the main array can also be an array. The idea is that, there is another array within an array and each element in the sub array can also be an array and so on. So, it is complicated.
- But you can create a family array, for example. So, in families there is an array of Gupta, so, the Gupta family is Raju, Sita and Payal, so, there is a Raju Gupta, Sita Gupta, and Payal Gupta. So, that is one array. So, the family is an array in which the first one, first array, let us call it $family [0] [Gupta]. And, that is Raju, Sita, Payal. So, that is 0 Gupta, 0 will be Raju Gupta, 1 will be Sita Gupta, 2 will be Payal. And, the same way there is a family is 1, actually family is 1, 0. There is a Singh also here. And, there is only one, that is Shanti. And then, there is a third one, which is called Brown. So, you can have multi-dimensional arrays. On these multi-dimensional arrays, I would request you guys to get into it, once you become very, very confident with the Associative array, only then get into this. Otherwise, it can be slightly confusing for you.

So, with this, we actually come to the end of the initial exposure to PHP and upgrade of exposure to what PHP is all about. I recommend you guys to go to W3schools assignment, as

I mentioned at the beginning of the co course and look at the PHP tutorial. It is a very quick way for you to try and learn PHP without actually having a PHP installed on your computer. So, W3schools is once more recommended for the PHP tutorial, and please go through it.

So, I am leaving the rest of the stuff for you guys to read, learn, practice, and then, we will now quickly see how to capture the forms from the data from the forms using PHP or how to integrate HTML form and PHP together, and then how to integrate DBMS with PHP together and then, tie it up all last part of the course at this point. So, thank you very much for your patience, listening and we will continue it in the next class. Thank you.