

## Chapter 1 : PHP: Function and Method listing - Manual

*C# question: I need to search backwards on a string using LastIndexOf() The string in question is "South Dakota " Need to split the state name from the system indicator.*

Module A Module is a collection of methods and constants. The methods in a module may be instance methods or module methods. Instance methods appear as methods in a class when the module is included, module methods do not. Conversely, module methods may be called without creating an encapsulating object, while instance methods may not. In the descriptions that follow, the parameter `sym` refers to a symbol, which is either a quoted string or a Symbol such as: This list includes the names of all modules and classes defined in the global scope. Think of the relationship in terms of the class definition: Typically, this method is overridden in descendant classes to provide class-specific meaning. This is used by Hash to test members for equality. For objects of class Object, `eq!`? Subclasses normally continue this tradition by aliasing `eq!`? Of limited use for modules, but can be used in case statements to classify objects by class. This can be used to add methods to a class. The optional `filename` and `lineno` parameters set the text for error messages. The method defined in the block will belong to the receiver. Any arguments passed to the method will be passed to the block. This can be used if the block needs to access instance variables. String arguments are converted to symbols. The part of the variable name should be included for regular class variables. If the class variable name is passed as a string, that string is converted to a symbol. This includes the names of class variables in any included modules, unless the `inherit` parameter is set to false. If `inherit` is false, the lookup only checks the constants in the receiver: If `inherit` is set, the lookup will also search the ancestors and Object if `mod` is a Module. The value of the constant is returned if a definition is found, otherwise a NameError is raised. It is passed a symbol for the undefined constant, and returns a value to be used for that constant. The following code is an example of the same: If found, it returns the loaded class. It therefore implements an autoload feature similar to Kernel `autoload` and `autoload`. Creates a new constant if no constant with the given name previously existed. This includes the names of constants in any included modules example at start of section , unless the `inherit` parameter is set to false. The implementation makes no guarantees about the order in which the constants are yielded. This method returns self. For a module, these are the public and protected methods; for a class, they are the instance not singleton methods. If the optional parameter is false, the methods of any ancestors are not included. Public and protected methods are matched.

### Chapter 2 : Sample: Smalltalk Methods Initially defined in Our Smalltalk.

*Definition of index method: The appraisal method of estimating building costs by multiplying the original cost of the property by a percentage factor to.*

After an attribute is associated with a program entity, the attribute can be queried at run time by using a technique called reflection. For more information, see Reflection C. Attributes have the following properties: Attributes add metadata to your program. Metadata is information about the types defined in a program. NET assemblies contain a specified set of metadata that describes the types and type members defined in the assembly. You can add custom attributes to specify any additional information that is required. For more information, see, Creating Custom Attributes C. You can apply one or more attributes to entire assemblies, modules, or smaller program elements such as classes and properties. Attributes can accept arguments in the same way as methods and properties. Your program can examine its own metadata or the metadata in other programs by using reflection. Using attributes Attributes can be placed on most any declaration, though a specific attribute might restrict the types of declarations on which it is valid. In C , you specify an attribute by placing the name of the attribute enclosed in square brackets [] above the declaration of the entity to which it applies. In this example, the SerializableAttribute attribute is used to apply a specific characteristic to a class: An example of such a multiuse attribute is ConditionalAttribute: However, you do not need to specify the attribute suffix when using attributes in code. Attribute parameters Many attributes have parameters, which can be positional, unnamed, or named. Any positional parameters must be specified in a certain order and cannot be omitted. Named parameters are optional and can be specified in any order. Positional parameters are specified first. For example, these three attributes are equivalent: In this case, both named parameters default to false, so they can be omitted. Positional parameters correspond to the parameters of the attribute constructor. Named or optional parameters correspond to either properties or fields of the attribute. Attribute targets The target of an attribute is the entity to which the attribute applies. For example, an attribute may apply to a class, a particular method, or an entire assembly. By default, an attribute applies to the element that it precedes. But you can also explicitly identify, for example, whether an attribute is applied to a method, or to its parameter, or to its return value. To explicitly identify an attribute target, use the following syntax:

**Chapter 3 : Pavement Condition Index - Wikipedia**

*Each time a method is invoked, the system stores parameters and local variables in an area of memory, known as \_\_\_\_\_, which stores elements in last-in first-out fashion. a stack Variables defined inside a method are called \_\_\_\_\_.*

Regions may be defined geographically e. The concept may be extended well beyond an exchange. The Wilshire Index, the original total market index, represents the stocks of nearly every publicly traded company in the United States , including all U. Other indices may track companies of a certain size, a certain type of management, or even more specialized criteria – one index published by Linux Weekly News tracks stocks of companies that sell products and services based on the Linux operating environment. The difference between the full capitalization, float-adjusted, and equal weight versions is in how index components are weighted. Thus, price movement of even a single security will heavily influence the value of the index even though the dollar shift is less significant in a relatively highly valued issue, and moreover ignoring the relative size of the company as a whole. Thus, a relatively small shift in the price of a large company will heavily influence the value of the index. Traditionally, capitalization- or share-weighted indices all had a full weighting, i. Recently, many of them have changed to a float -adjusted weighting which helps indexing. An equal-weighted index is one in which all components are assigned the same value. It is similar to a capitalization weighting with one main difference: For these two indexes, a score is calculated for every stock, be it their growth score or the value score a stock cannot be both and accordingly they are weighted for the index. This then gives the average return for all investors; if some investors do worse, other investors must do better excluding costs. This considers risk and return and does not consider weights relative to the entire market. This may result in overweighting assets such as value or small-cap stocks, if they are believed to have a better return for risk profile. These investors believe that they can get a better result because other investors are not very good. The capital asset pricing model says that all investors are highly intelligent, and it is impossible to do better than the market portfolio, the capitalization-weighted portfolio of all assets. However, empirical tests conclude that market indices are not efficient. The practical conclusion is that using capitalization-weighted portfolios is not necessarily the optimal method. As a consequence, capitalization-weighting has been subject to severe criticism see e. Haugen and Baker , Amenc, Goltz, and Le Sourd , or Hsu , pointing out that the mechanics of capitalization-weighting lead to trend-following strategies that provide an inefficient risk-return trade-off. Also, while capitalization-weighting is the standard in equity index construction, different weighting schemes exist. Scorecard," which measures the performance of indices versus actively managed mutual funds, finds the vast majority of actively managed mutual funds underperform their benchmarks. Indices are also a common basis for a related type of investment, the exchange-traded fund or ETF. Unlike an index fund, which is priced daily, an ETF is priced continuously, is optionable, and can be sold short. Ethical stock market indices[ edit ] A notable specialized index type is those for ethical investing indices that include only those companies satisfying ecological or social criteria, e. Ethical indices have a particular interest in mechanical criteria, seeking to avoid accusations of ideological bias in selection, and have pioneered techniques for inclusion and exclusion of stocks based on complex criteria. Another means of mechanical selection is mark-to-future methods that exploit scenarios produced by multiple analysts weighted according to probability, to determine which stocks have become too risky to hold in the index of concern. Critics of such initiatives argue that many firms satisfy mechanical "ethical criteria", e. Indeed, the seeming "seal of approval" of an ethical index may put investors more at ease, enabling scams. One response to these criticisms is that trust in the corporate management, index criteria, fund or index manager, and securities regulator, can never be replaced by mechanical means, so " market transparency " and " disclosure " are the only long-term-effective paths to fair markets. From a financial perspective, it is not obvious whether ethical indices or ethical funds will out-perform their more conventional counterparts. Theory might suggest that returns would be lower since the investible universe is artificially reduced and with it portfolio efficiency. On the other hand, companies with good social performances might be better run, have more committed workers and customers, and be less likely to suffer reputational damage from incidents oil

spillages, industrial tribunals, etc. Sharpe Indexing Achievement Awards are presented annually in order to recognize the most important contributions to the indexing industry over the preceding year.

**Chapter 4 : A Transformation of Accelerated Double Step Size Method for Unconstrained Optimization**

*Since it says that the method is undefined for nil (as described above) it tells us that on the index page, @users is nil (probably not what it is intended to be!). So it looks like the problem rests in your definition of @users.*

I try to make the parts that are not verbatim to be descriptive of the expected use. We will use these conventions shortly in the discussion of function syntax, and will continue to use the conventions throughout the tutorial. We can make Python display the song. Read, and run if you like, the example program birthday1. In Python we can also give a name like happyBirthdayEmily, and associate the name with whole song by using a function definition. We use the Python def keyword, short for define. The heading contains def, the name of the function, parentheses, and finally a colon. The remaining lines form the function body and are indented by a consistent amount. The exact amount is not important to the interpreter, though 2 or 4 spaces are common conventions. The whole definition does just that: This is our first example of altering the order of execution of statements from the normal sequential order. Note The statements in the function definition are not executed as Python first passes over the lines. The code above is in example file birthday2. Load it in Idle and execute it from there. Nothing should happen visibly. This is just like defining a variable: Python just remembers the function definition for future reference. After Idle finished executing a program, however, its version of the Shell remembers function definitions from the program. When you give the Shell an identifier, it tells you its value. Above, without parentheses, it identifies the function code as the value and gives a location in memory of the code. Now try the name in the Idle Shell with parentheses added: Python goes back and looks up the definition, and only then, executes the code inside the function definition. The term for this action is a function call or function invocation. Note In the function call there is no def, but there is the function name followed by parentheses. This is not true if you run a program by selecting it directly in the operating system. Look at the example program birthday3. See it just adds two more lines, not indented. Can you guess what it does? Python starts from the top, reading and remembering the definition. The definition ends where the indentation ends. The code also shows a blank line there, but that is only for humans, to emphasize the end of the definition. The code of the function is executed for the first time, printing out the song. End of line 9: Back from the function call. The function is executed again, printing out the song again. End of line Back from the function call, but at this point there is nothing more in the program, and execution stops. Functions alter execution order in several ways: Think how to do that before going on Guess what happens, and then try it: They are the only lines executed directly. The calls to the functions happen to be in the same order as their definitions, but that is arbitrary. If the last two lines were swapped, the order of operations would change. Do swap the last two lines so they appear as below, and see what happens when you execute the program: It is a good convention to have the main action of a program be in a function for easy reference. The example program birthday5. Do you see that this version accomplishes the same thing as the last version? The final line is the only one directly executed, and it calls the code in main, which in turn calls the code in the other two functions. Detailed order of execution: Definitions are read and remembered Line The only statement outside definitions, is executed directly. This location is remembered as main is executed. Start on main Line This location is remembered as execution jumps to happyBirthdayEmily Lines are executed and Emily is sung to. Return to the end of Line Back from happyBirthdayEmily function call Line Now happyBirthdayAndre is called as this location is remembered. Sing to Andre Return to the end of line Back from happyBirthdayAndre function call, done with main Return to the end of line Back from main; at the end of the program There is one practical difference from the previous version. After execution, if we want to give another round of Happy Birthday to both persons, we only need to enter one further call in the Shell to: What should happen now? The lines outside any function definition not indented are executed in order of appearance. Give a meaningful name to the function. Have the program end by calling the function three times, so the poem or verse is repeated three times. You had the power of abstraction. With examples like the versions for Emily and Andre, you could figure out what change to make it so the song could be sung to Maria! Unfortunately, Python is not that smart. It needs explicit rules. If you needed to explain explicitly to

someone how Happy Birthday worked in general, rather than just by example, you might say something like this: Python works something like that, but with its own syntax. This is just like the association with a variable name in Python. The function definition indicates that the variable name `person` will be used inside the function by inserting it between the parentheses of the definition. Read and then run example program `birthday6`. This variable name is a placeholder for the real name of the person being sung to. The last two lines of the program, again, are the only ones outside of definitions, so they are the only ones executed directly. There is now an actual name between the parentheses in the function calls. The value between the parentheses here in the function call is referred to as an argument or actual parameter of the function call. The argument supplies the actual data to be used in the function execution. When the call is made, Python does this by associating the formal parameter name `person` with the actual parameter data, as in an assignment statement. We say the actual parameter value is passed to the function. Definition remembered Line 9: End of line 9 after returning from the function call Line End of line 10 after returning from the function call, and the program is over. Note Be sure you completely understand `birthday6`. It illustrates extremely important ideas that many people miss the first time! It is essential to understand the difference between Defining a function lines with the `def` heading including formal parameter names, where the code is merely instructions to be remembered, not acted on immediately. Calling a function with actual parameter values to be substituted for the formal parameters and have the function code actually run when the instruction containing the call is run. Also note that the function can be called multiple times with different expressions as the actual parameters line 9 and again in line The beauty of this system is that the same function definition can be used for a call with a different actual parameter, and then have a different effect. The value of the formal parameter `person` is used in the third line of `happyBirthday`, to put in whatever actual parameter value was given. Note This is the power of abstraction. It is one application of the most important principal in programming. Rather than have a number of separately coded parts with only slight variations, see where it is appropriate to combine them using a function whose parameters refer to the parts that are different in different situations. Then the code is written to be simultaneously appropriate for the separate specific situations, with the substitutions of the right parameter values. You can go back to having a main function again, and everything works. We can combine function parameters with user input, and have the program be able to print Happy Birthday for anyone. There are more than one way to get information into a function: Have a value passed in through a parameter from line 10 to line 3.

**Chapter 5 : Weighted Methods per Class - Maisqual Wiki**

*Searches for an element that matches the conditions defined by the specified predicate, and returns the zero-based index of the first occurrence within the range of elements in the List that starts at the specified index and contains the specified number of elements.*

**Arrays** An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed. You have seen an example of arrays already, in the main method of the "Hello World! This section discusses arrays in greater detail. An array of 10 elements. Each item in an array is called an element, and each element is accessed by its numerical index. As shown in the preceding illustration, numbering begins with 0. The 9th element, for example, would therefore be accessed at index 8. The following program, `ArrayDemo` , creates an array of integers, puts some values in the array, and prints each value to standard output. Element at index 0: However, the example clearly illustrates the array syntax. You will learn about the various looping constructs `for`, `while`, and `do-while` in the Control Flow section. **Declaring a Variable to Refer to an Array** The preceding program declares an array named `anArray` with the following line of code: The size of the array is not part of its type which is why the brackets are empty. As with variables of other types, the declaration does not actually create an array; it simply tells the compiler that this variable will hold an array of the specified type. Similarly, you can declare arrays of other types: **Creating, Initializing, and Accessing an Array** One way to create an array is with the `new` operator. The next statement in the `ArrayDemo` program allocates an array with enough memory for 10 integer elements and assigns the array to the `anArray` variable. Variable `anArray` may not have been initialized. The next few lines assign values to each element of the array: You can also declare an array of arrays also known as a multidimensional array by using two or more sets of brackets, such as `String[][]` names. Each element, therefore, must be accessed by a corresponding number of index values. In the Java programming language, a multidimensional array is an array whose components are themselves arrays. This is unlike arrays in C or Fortran. A consequence of this is that the rows are allowed to vary in length, as shown in the following `MultiDimArrayDemo` program: **Jones** Finally, you can use the built-in `length` property to determine the size of any array. The three `int` arguments specify the starting position in the source array, the starting position in the destination array, and the number of array elements to copy. The following program, `ArrayCopyDemo` , declares an array of `char` elements, spelling the word "decaffeinated. Java SE provides methods to perform some of the most common manipulations related to arrays. For instance, the `ArrayCopyDemo` example uses the `arraycopy` method of the `System` class instead of manually iterating through the elements of the source array and placing each one into the destination array. This is performed behind the scenes, enabling the developer to use just one line of code to call the method. For your convenience, Java SE provides several methods for performing array manipulations common tasks, such as copying, sorting and searching arrays in the java. For instance, the previous example can be modified to use the `copyOfRange` method of the `java. Arrays` class, as you can see in the `ArrayCopyOfDemo` example. The difference is that using the `copyOfRange` method does not require you to create the destination array before calling the method, because the destination array is returned by the method: Note that the second parameter of the `copyOfRange` method is the initial index of the range to be copied, inclusively, while the third parameter is the final index of the range to be copied, exclusively. In this example, the range to be copied does not include the array element at index 9 which contains the character `a`. Some other useful operations provided by methods in the java. Searching an array for a specific value to get the index at which it is placed the `binarySearch` method. Comparing two arrays to determine if they are equal or not the `equals` method. Filling an array to place a specific value at each index the `fill` method. Sorting an array into ascending order. This can be done either sequentially, using the `sort` method, or concurrently, using the `parallelSort` method introduced in Java SE 8. Parallel sorting of large arrays on multiprocessor systems is faster than sequential array sorting.

### Chapter 6 : Class: Module (Ruby )

*The indexOf() method returns the first index at which a given element can be found in the array, or -1 if it is not present. The index to start the search at. If the index is greater than or equal to the array's length, -1 is returned, which means the array will not be searched. If the provided.*

By Florian Rapp 5 jQuery. In addition to this function, jQuery provides a helper function with the same name that can be called without having previously selected or created DOM elements. A possible output is: In this case the object to loop over is given as the first argument. In this example I show how to loop over an array: However, since the context will always be an object, primitive values have to be wrapped. Therefore, strict equality between the value and the context may not be given. The first argument is the current index, which is either a number for arrays or string for objects. The first example selects all the a elements in the page and outputs their href attribute. The reason is that jQuery is in fact just a wrapper around an array of DOM elements. By using jQuery each this array is iterated in the same way as an ordinary array would be. An array features numeric indices, hence we obtain numbers starting from 0 and going up to  $N - 1$ , where  $N$  is the number of elements in the array. We handle the nested structure with a nested call to each. The outer call handles the array of the variable JSON, the inner call handles the objects. In this example each object has only one key, however, in general any number could be attacked with the provided code. These are just parameters which help determine on which DOM element we are currently iterating. Furthermore, in this scenario we can also use the more convenient each method. We can write it like this: After an index-dependent delay 0, , , milliseconds we fade out the element. Florian regularly gives talks at conferences or user groups. You can find his blog at [florian-rapp](#).

### Chapter 7 : Defining Adult Overweight and Obesity | Overweight & Obesity | CDC

*Use the selectionSort method presented in this section to answer this question. Assume list is { , , , }, what is the content of list after the first iteration of the outer loop in the method?*

### Chapter 8 : Attributes (C#) | Microsoft Docs

*Guidance about methods and approaches to achieve de-identification in accordance with the Health Insurance Portability and Accountability Act of*

### Chapter 9 : 5 calendrierdelascience.com() Function Examples – SitePoint

*The method defined in this way is comparable with its double step size forerunner, ADSS method, as well as with the single step size accelerated gradient descent SM method which is defined in a classical manner.*