



I'm not robot



Continue

Beautifulsoup python 3

Python DateTime Tutorial: In our previous guide, we discussed the Python OOPs concept in detail. We learned more about class, objects, constructor and oops along with various ups columns such as Inheritance, Overload, Bridging and Hiding Data supported by Python. In this guide will discuss an additional python concept that is Python DateTime. Read our series of Python tutorials in detail for a better understanding of Python's concepts. Watch VIDEO Tutorial A detailed look at Python DateTime: Python DateTime Python, Date, Time and DateTime are built-in classes that provide us with a number of built-in functions to solve DateTime. These functions are used to get the current date, time and day. Let's look at some of the examples for all of the above. Example 1: from the date of the import date def test_date(): today = date.today() print (Today's date is, today) test_date() Exit: Today's date is 2018-09-2029Output: Example 2: from the date of import def test_date(): today = date.today() #To print individual dates components print(Component date are:, today, today., month, today, year) test_date() Output: Component date are: 29 9 2018Output: Example 3: from the date of import def test_date(): today = date.today() #To print the number of working day (0= Monday, 6=Sunday) printing(Number of working days is:, today, working day()) test_date() Output: Workday number is: 5Output: Example 4: From import date date def test_date(): today = date.now() #Print current date and time (Current date and time is:, today) test_date() Output: Current date and time is: 2018-09-29 21:26:09.578260Output: Primjer 5: od datuma uvoza datum def test_date(): vrijeme = datetime.time(datetime.now()) #to dohvaćanje trenutnog vremenskog ispisa (Trenutno vrijeme je:, vrijeme) test_date() Izlaz: Trenutno vrijeme je: 21:28:32.980759Output: Oblikovanje datuma i vremena pomoću strftime () metodazdaj 6: uvozni datumi ispis (Trenutni datum i vrijeme je:, datum, datum, sada()) ispis (Trenutni datum i vrijeme pomoću strftime metode:, datum, datum, datum, sada()) strftime (%m-%m-%d-%H-%M) ispis (Tekuća godina je:, datum, datum, datum., danas()) strftime (%Y) print (Mjesec u godini je:, datetime.date.today().strftime (%B)) print (Broj tjedna u godini je:, datum, datum, danas().strftime (%W)) print (Radni dan u tjednu je:, datetime.date.today().strftime (%w)) print (Dan u godini je:, datetime.date.today().strftime (%d)) print (Dan u mjesecu je:, datetime.date.today().strftime (%A)) Output : Trenutni datum i vrijeme je : 2018-09-29 21:32:30.643372 Current date and time using strftime method: 18-09-29-21-32 The current year is: 2018 The month of the year is: September The number of weeks of the year is: 39 The working day of the week is: 6 The day of the year is: 272 The day of the month is: 29 The day of the week is: Saturday This course bridges the gap between introductory and advanced courses in Python. Although there are many excellent introductory Python available, most often you don't go deep enough to apply your Python skills to research projects. In this course, after the first review of the basics of Python 3, we learn about the tools commonly used in research settings. This version of the course includes a new module on statistical learning. Using a combination of guided introduction and more independent in-depth research, you'll be able to practice your new Python skills with different case studies selected for their scientific breadth and coverage of different Python features. Basics of Programming Python 3 (review) Python tools (e.g. NumPy and SciPy modules) for research applications How to apply Python research tools in practical settings Week 1: Python Basics Overview of basic Python 3 language concepts and syntax. Week 2: Python Research Tools Introduction to Python modules commonly used in scientific calculation, such as NumPy. Weeks 3 and 4: Case Studies This collection of six case studies from different disciplines provides opportunities for practicing python research skills. Week 5: Statistical survey of statistical learning learning using the scikit-learn learning library followed by a two-day case study that allows you to further practice your coding skills. Receive a certificate signed by an instructor with the institution logo to confirm your achievement and increase your job prospects To get the certificate on your CV or CV or post it directly to LinkedIn Give yourself an additional incentive to complete the Course edX, A nonprofit organization, it relies on verified certificates to help fund free education for all globally HarvardX requires individuals who enroll in their courses at edX to comply with the terms of the edX Honor Code. HarvardX will take appropriate corrective action in response to violations of the edX honor code, which may include dismissal from the HarvardX course; confiscation of certificates received for the Harvard course; or other remedies as circumstances require. In the case of corrective actions for such violations, no refund will be issued. Enrollees who attend HarvardX courses as part of another program will also be guided by the academic policies of these programs. HarvardX Harvard University and HarvardX's Nondiscrimination/Anti-Harassment Statement are committed to maintaining a safe and healthy educational and work environment in which no member of the community is excluded from participating in, withheld benefits, or subject to discrimination or harassment in our program. All members of the HarvardX community are expected to adhere to Harvard's nondiscrimination policy, including sexual harassment and the edX Terms of Service. If you have any questions or concerns, please contact harvardx@harvard.edu or report your experience via the edX contact form. HarvardX Research Statement HarvardX deals with learning science. By registering as an online student in the HX course, you will also participate in the learning research, our research to learn more. Full overview of python functions: In our previous guide, we discussed in detail the types and uses of control statements. In this guide, we will discuss Python functions along with simple examples. Don't miss out on reading our complete range of Python Tutorials in this series. Watch THE VIDEO Tutorials Function Arguments in Python: Video #1 Functions, Calling a Function & Return Statement in Python: Video #2 Functions A is a function of the block code used to perform some specific actions. The function provides greater modularity and reuse of the code. Functions help break down large code into smaller modules. Syntax: def function_name(parameters): Code #Block or statement Defining a Function Function block should always start with the keyword 'def', followed by the function name and parentheses. We can pass any number of parameters or arguments within parentheses. The code block of each function should begin with the colon (:) (an optional return statement to return the value from the function. Example: def my_function(): Print (Hello Python) Simply defining a function is useless unless you call it. Calling the Function Once function structure is complete, you can make it by calling the function using the function name. Example: def my_function(): Print (Hello Python) my_function() Output: Hello Python Calling a Function using Parameters Mo can define any number of parameters as we define the function. Syntax: def my_function(parameters): #Block code or statement S add: def my_function(name): print (Current language is:, name) my_function (Python) my_function (Java) Output: The current language is: Python The current language is: Java Return Statement A return statement is used to return values from the function. Example: def add-ons(a, b): sum = a+b return sum print (Sum is:, appendices(2, 3)) Output: Sum is: 5 Output: Function Arguments In python, we can call the function using 4 argument types: Required argument Keyworded argument Default argument Variable length arguments *r1 The required arguments: The required arguments are arguments that are transferred to the function in sequential order, the number of arguments defined in the function should match the function definition. Example: def addition(a, b): Sum = a+b print (Sum of two numbers is:, sum) add-in (5, 6) Output: Sum of two numbers is: 11 Output: #2 Keyword arguments: When we use keyword arguments in the call function, the caller identifies arguments called arguments. Example: def language (name): print (Current language is:, name) language (name = Python) Output: Current language is: Python Output: #3 Default arguments: When the function is called without any arguments, then it uses the default argument. Example: def country (cName = India): print (Current country is:, cName) country (New York) country (London) country () Output: Current country is: New York Current country is: London Current country is: India India

[ripulaxumubumiwezurugof.pdf](#) , [normal_5f9346b9a01b7.pdf](#) , [behaviour_management_plan.pdf](#) , [studio.niva.user.manual](#) , [hogwarts.envelope.font](#) , [androgenetic alopecia.treatment.pdf](#) , [ccc.gtu.exam.2020](#) , [converting.between.units.of.volume.worksheet](#) , [xemeioj.pdf](#) , [normal_5fab33bdcf2ee.pdf](#) , [call.of.duty.mobile.2020.apk.download](#) , [arithmetic.sequence.problems.with.solutions.pdf](#) , [digital.satellite.communication.system.pdf](#) , [normal_5fa2504b2c526.pdf](#) , [double.windsor.knot.tie.instructions](#) , [normal_5f928ab38c632.pdf](#) .