**Test Task for Tutor Python**

**Python**

**What are the main differents between Python2 and Python3?**

* Print statement.   
  In Python 3 the print() statement is a function and therefore you have to enclose in parentheses what you want to print, while with Python 2 the parentheses are not necessary. Example:  
  Python 2  
  >>> print ‘Hello World’  
  Python 3

>>> print (‘Hello World’)

* Division of Integers.  
  In Python 2, division by integers is another integer, and to get a result with decimals either the numerator of denominator must also have at least one decimal place. Example:

>>>1/2

0

>>>1.0/2

0.5

In Python 3:

>>>1/2

0.5  
>>>1.0/2  
0.5  
>>1//2

0

* \_future\_ Module.  
  This module provides backward compatibility of unsupported commands between different versions. We can use it to adapt Python 2 to Python 3 syntax.
* Type comparison.  
   Python 2 allows the comparison between objects of different types, however Python 3 is restrictive in this aspect giving us a TypeError type exception.
* Input() function.  
  In Python 3 the raw\_input function has been removed, and its behavior is taken over by the input() function. This facilitates the treatment of the data entered by keyboard since we know in advance that they will always be Strings.

**What happens if you import the same library into the code twice?**

The module is only loaded the first time the import statement is executed and there is no performance loss by importing it again.

**What is cyclic import and how can you avoid it?**

A cyclic import is an import which imports another module and that module imports (possibly indirectly) the module which contains the import statement. Cyclic imports indicate that two modules are circularly dependent. This means that the modules cannot be tested independently, and it makes it harder to understand the architecture of the system.  
You can avoid it putting all imports in a central module, instead to import A into B, and B into A; you can create an C file where you can import A and B.

**What is a context manager?**

A context manager usually takes care of setting up some resource, e.g. opening a connection, and automatically handles the clean up when we are done with it. Example:

With open (“test.txt”) as f:  
 data = f.read()  
When creating context managers using classes, user need to ensure that the class has the methods: *\_\_enter\_\_()* and *\_\_exit\_\_()*. The \_\_enter\_\_() returns the resource that needs to be managed and the \_\_exit\_\_() does not return anything but performs the cleanup operations.

**Describe how a list data structure works?**A list is a mutable, or changeable, ordered sequence of elements. Each element or value that is inside of a list is called an item. Just as strings are defined as characters between quotes, lists are defined by having values between square brackets [ ].

**Algorithms**

**Which data type in Python should I choose for the items container so that the operation of searching for an item is asymptotically the most efficient?**

  - A set

  - A list

  - Sorted List

  - A single-linked list

  - It doesn't matter, since we'll write for items in items: **(Correct)**

  - tuple

**Estimate the difficulty of quick sorting in the worst case scenario.**

  - O(N^2)

  - O(N logN)

  - O(log(logN))

The worst case time complexity of a typical implementation of [QuickSort](http://geeksquiz.com/quick-sort/)is O(N^2). The worst case occurs when the picked pivot is always an extreme (smallest or largest) element. This happens when input array is sorted or reverse sorted and either first or last element is picked as pivot.

**Estimate the complexity of the operation of searching for an element in the search tree in the worst case scenario.**

  - O(logN) **(Correct)**

  - 0(N logN)

  - O(N^2)

  -There are no correct answers

**What is a "pyramid"**

The pyramid data structure defines the relationship between the different layers of data required to build a pyramidal structure. In finite element data, you can consider the element grid as a collection of vertices (points), edges (lines), faces (polygons), elements (three-dimensional cells), objects (collections of 3-D elements), assemblies (collections of objects), and so on. You can fit these together to make an object from faces, bounded by edges, which are in turn delimited by vertices.  
Because Pyramid combines several data types, it is an extremely flexible and powerful structure; witness its ability to handle both finite element data and molecular data. The flexibility of the Pyramid data type also makes it somewhat difficult to use properly.

**What features should a binary tree have in order to be a search tree?**

A binary search tree have this specific characteristics:  
 - Each node has a maximum of up to two children.  
 - For each node, the values of its left descendent nodes are less than that of the current node, which in turn is less than the right descendent nodes (if any).

**Django**

**How can I get the value of the username variable in the controller if it is passed from the page by the POST method?**

    - request.POST.get('username’)

    - request.POST['username'] **(Correct)**

    - request.POST.username

    - request.POST('username')

\* Which command runs the development web server in Django?

    - python manage.py runserver **(Correct)**

    - python manage.py devserver

    - python run server

    - python devserver run

**What arguments in this code are passed to the view function?**

    path('/example/<int:id>/', example\_view)

    - Named int argument that has an id type

    - Named id argument that has an int type

    - Positioned int argument that has an id type

    - Positioned id argument that has and int type **(Correct)**

**You need to write a controller that returns data from the database in the form of an object. On the basis of which class would you create it?**

    - ListView

    - DetailView

    - RetrieveView

    - GetView **(Correct)**

**You have changed the models in a project. What commands do you need to run to commit changes and start migrations?**python manage.py makemigrations

**What is middleware?**In Django, middleware is a lightweight plugin that processes during request and response execution. Middleware is used to perform a function in the application. The functions can be a security, session, csrf protection, authentication etc.

**SQL**

**What types of JOIN queries in SQL do you know? Describe them and specify the differences beetween them.**

**JOIN is an SQL clause used to query and access data from multiple tables, based on logical relationships between those tables.  
Differents types of Joins:  
 INNER JOIN  
 FULL OUTER JOIN  
 LEFT OUTER JOIN  
 RIGHT OUTER JOIN  
 SELF JOIN  
 CROSS JOIN**

**INNER JOIN statement returns only those records or rows that have matching values and is used to retrieve data that appears in both tables. If you use an SQL OUTER JOIN, it will retrieve not only the matching rows but also the unmatched rows as well.  
The LEFT OUTER JOIN gives the output of the matching rows between both tables and the left table too. The RIGHT OUTER JOIN is similar, shows the matching between tables and the right table.  
SELF JOIN statement allow you to join a table to itself. This implies that each row of the table is combined with itself and with every other row of the table.  
CROSS JOIN return all combinations of rows from each table.**

**There are two tables: A and B. The first table has 4 columns and 10 records, and the second one has 5 columns and 8 records. How many rows and columns will be output with the**

**select \* from A,B query, and why?**

It will have 9 columns and 80 rows. The SQL engine perfoms the cartesian product of both sets, combining all the records of Table A with all the records of Table B, so that each result row is one of the possible combinations. Therefore the number of resulting rows will be equal to the number of records in Table A multiplied by the number of records in Table B.

**Indicate the main differences between SQL and NOSQL databases. Give examples how both of them can be used and explain your choice**SQL and NoSQL differ in whether they are relational (SQL) or non-relational (NoSQL), whether their schemas are predefined or dynamic, how they scale, the type of data they include and whether they are more fit for multi-row transactions or unstructured data.

SQL is still widely used for querying relational databases, where data is stored in rows and tables that are linked in various ways. One table record may link to one other or to many others, or many table records may be related to many records in another table. These relational databases, which offer fast data storage and recovery, can handle great amounts of data and complex SQL queries.  
[NoSQL](https://www.ibm.com/cloud/learn/nosql-databases) is a non-relational database, meaning it allows different structures than a SQL database (not rows and columns) and more flexibility to use a format that best fits the data.  
Examples:  
SQL Databases: MySQL, PostgreSQL, Microsoft SQL Server  
NoSQL Databases: MongoDB, Redis, Elasticsearch

**General questions**

**You need to write Python code that will send user's comments to a certain forum. Which request method would you choose for this - GET, POST, of PUT? Explain you choice.**POST Method. Because it is the method to send data to a server to create/update a resource. The POST request are never cached, do not remain in the browser history and have no restrictions on data length.

**Explain which is better: inheritance of composition. Why?**Both composition and inheritance promote code reuse through different approaches.  
-Inheritance is tightly coupled whereas composition is loosely coupled.  
-There is no access control in inheritance whereas access can be restricted in composition  
-Composition provides flexibility in invocation of methods that is useful with multiple subclass scenario.  
-One more benefit of composition over inheritance is testing scope. Unit testing is easy in composition because we know what all methods we are using from another class. We can mock it up for testing whereas in inheritance we depend heavily on superclass and don't know what all methods of superclass will be used.  
For all this reasons composition is a better choice.

**Explain how authentication differs from authorization**

In simple words, authentication verifies the identity of a user or service, and authorization determines their access rights.

**What is versioning used for when creating REST API services?**API versioning is the practice of managing changes to an API and ensuring that these changes are made without disrupting clients. A good API versioning strategy clearly communicates the changes made and allows API consumers to decide when to upgrade to the latest version at their own pace.

**What HTTP methods do you know?**

POST, GET, PUT, PATCH and DELETE

**Suggest several options for scaling your service**Splitting services: splitting monolith software into smaller ones (Microservices)  
Horizontal scaling  
Separate databases  
Using cloud services

**What ways do you know for changing file access rights in Linux?**- chmod +rwx filename to add permissions.  
- chmod -rwx directoryname to remove permissions.  
- chmod +x filename to allow executable permissions.  
- chmod -wx filename to take out write and executable permissions

**List the main differences between the process and the flow?**A Process is a sequence of tasks, The flow is the way in which this sequence is most efficient and most productive. A Process exists naturally and intuitively, the Workflow is planned, analyzed, designed and implemented with a defined intention.

**What ways of iteration between processes do you know?**

Planning, analysis, implementation and evaluation.

**What types of DNS records do you know?**- Address mapping records  
- AAAA  
- Mail exchanger record  
- CERT record  
- SRV record  
- SOA record  
- TXT record  
- PTR record

**0 STDIN, 1 STDOUT, 2 STDERR -- what does it mean?**

stdin, stdout, and stderr are three data streams created when you launch a Linux command. You can use them to tell if your scripts are being piped or redirected.

**What is the name of the DNS record for IPv6?**  
The name is AAAA.

**Others**

**Why do you want to become a Tutor in Practicum?**Because I like being able to help other people to start in the world of programming. I believe that I have the necessary knowledge and experience to do so.

**What competencies do you think a good tutor in your profession should have? Explain your opinion. Give examples from your own experience**Empathy, patience, dedication to service, active listening, proactivity.

In my experience as a python and backend programming tutor, students need a guide, a mentor, someone who listens to them and helps them to understand the logic of programming more easily, supports them in the face of frustration and guides them successfully so that They can learn as much as possible and are excellent professionals

**Where did you study back-end development and how did you get into this profession?**I started studying on my own taking courses on Udemy and FreeCodeCamp. Then I did an intensive Fullstack Programming Bootcamp and started my studies in Systems Engineering and Artificial Intelligence.

On the other hand, during my last 5 years I have been collaborating as a developer in a family business and currently working as a Fullstack Developer and Project Leader in a financial company.