## Python

**Advanced:**

1. Pickle usage OR Object serialization process in python
2. Regular Expressions
3. Decorators purpose and usage (with and without args passing to decorator)
4. Difference b/w range and xrange
5. Purpose and usage of Zip and sum
6. Difference b/w Map, Reduce and Filter
7. To check for an attribute in an object what is the method we need to use
   1. Ans: hasattr(obj, attr\_name) -> returns True if the attr\_name is the name of one of the obj’s attribute. Else return false.
8. Xml parsing using python
9. Subprocess (both synchronous and asynchronous)
10. @property decorator purpose
11. @staticmethod vs @classmethod vs instance method
12. \_\_repr\_\_(self) method purpose
13. \_\_new\_\_(cls) vs \_\_init\_\_(self)
14. Time complexity for any simple program (Ex: find second highest number from a list of values)
15. Xml passing for SOAP requests from python
16. Compile time and run time for python programs
17. Finding the duplicate elements inside a list

|  |
| --- |
| Import collections  X = [0,1,2,2,3,4,5,5]  Y = collections.counter(x)  Print y (output: [ { 2:2,5:2,0:1,1:1,3:1} ] )  Z = list(y) (returns unique list) |

**Basic:**

1. Usage of enumerate()
2. Formatting array in to json objects and vice versa.
3. Generators vs Iterators
4. Purpose and usage of With statement OR Necessity of content managers in python programming
5. Higher order function vs Lower order function
6. List comprehension vs regular loops
7. Package vs module
8. Slicing, Indexing and reversing strings and lists in python.
9. Lambda usage

## Django:

**Advanced:**

1. Declaring an attribute of a model as “read only”
2. Database Transaction Management (like save, commit, rollback)
3. Multiple databases (default, employee. Etc..) and manually selecting while selecting and saving the objects
4. Writing custom filters and tags
5. Writing custom decorators
6. Meta class purpose inside a model
7. Handling User Roles and permissions
8. Handling Sessions
9. Purpose and usage of a middleware? Asking to state some middlewares which were used?
10. Customizing the save() method of a model
11. Verbose\_name purpose OR customizing the model name in the DB
12. Django applications like south, pagination, REST framework, Django-LDAP etc..
13. How to return 404 error response from Django view?
14. How to provide “one to one” relation b/w models?

**Basic:**

1. Custom validations for form fields [ like password length should be morethan 8 ] OR custom validations in form
2. Adding select widget to a form field
3. Purpose and Using Q() inside a filter
4. Purpose of reverse() and resolve()
5. Purpose of a template context object OR how to determine user’s permissions in template
6. Effect of “syncdb” on a changed model (suppose you have added a new field to the existing model. If we run the “syncdb” command what will happen)
7. Purpose of “select\_related”
8. Way of getting count of records

## MySQL:

1. How many types of storage engines are available?
2. Diff. b/w myIsam and InnoDB
3. What is normalization? Explain level wise?
4. How many types of joins are available?
5. Advantages of Joins over sub queries.
6. 1st Normal form Vs 3rd Normal Form
7. Purpose of Indexing; Its advantages and disadvantages.
8. Designing the Database for a small scenario.