

# Співбесіда з Python. Запитання для Junior

## Загальне з Computer Science та Web Development:

1. Що таке інженерія і процес розробки загалом?
2. Які знаєте принципи програмування?
3. Чим відрізняються процедурна та об'єкто-орієнтована парадигми програмування?
4. Які основні принципи ООП (наслідування, інкапсуляція, поліморфізм)?
5. Що таке множинне наслідування?
6. Які є шість етапів розробки продукту в Software Development lifecycle і яка різниця між Agile і Kanban?
7. Які є методи HTTP-запитів та яка між ними різниця?
8. Як виглядають HTTP-request/response?
9. Що таке авторизація і як вона працює?
10. Що таке cookies?
11. Що таке вебвразливість?
12. Які знаєте класичні бази даних?
13. Як читати специфікацію в конкретній мові (наприклад, PEP8 у Python)?
14. Як відбувається взаємодія клієнта і сервера?
15. Які є підходи до проєктування API?
16. Як використовувати патерни програмування?
17. Що таке Acceptance Testing і навіщо його використовують?
18. Що таке модульні та інтеграційні тести, API-тести?
19. Як писати unit-тести?
20. Які є best practices у написанні автотестів?
21. Які базові команди системи контролю версій?
22. Як використовувати Git?
23. У чому різниця між хешуванням і шифруванням?

## Python:

24. Python — мова, що інтерпретується чи компілюється?
25. Які є змінні та незмінні типи даних?
26. Що таке область видимості змінних?
27. Що таке introspection?
28. Різниця між `is` і `==`?
29. Різниця між `__init__()` і `__new__()`?

- 30. У чому різниця між потоками та процесами?
- 31. Які є види імпорту?
- 32. Що таке клас, ітератор, генератор?
- 33. Що таке метаклас, змінна циклу?
- 34. У чому різниця між ітераторами та генераторами?
- 35. У чому різниця між `staticmethod` та `classmethod`?
- 36. Як працюють декоратори, контекстні менеджери?
- 37. Як працюють `dict comprehension`, `list comprehension` і `set comprehension`?
- 38. Чи можна використовувати кілька декораторів для однієї функції?
- 39. Чи можна створити декоратор з класу?
- 40. Які є основні популярні пакети (`requests`, `pytest`, etc)?
- 41. Що таке `lambda`-функції?
- 42. Що означає `*args`, `**kwargs` та як вони використовуються?
- 43. Що таке `exceptions`, `<try-except>`?
- 44. Що таке PEP (Python Enhancement Proposal), які з них знаєте (PEP 8, PEP 484)?
- 45. Напишіть `hello-world` сервіс, використовуючи один із фреймворків.
- 46. Які є типи даних і яка різниця між `list` і `tuple`, навіщо вони?
- 47. Як використовувати вбудовані колекції (`list`, `set`, `dictionary`)?
- 48. У чому полягає складність доступу до елементів `dict`?
- 49. Як створюється об'єкт в Python, для чого `__new__`, навіщо `__init__`?
- 50. Що знаєте з модуля `collections`, якими ще `built-in` модулями користувались?
- 51. Що таке шаблонізатор та як у ньому виконувати базові операції (об'єднувати ділянки шаблону, виводити дату, виводити дані з серверного боку)?
- 52. Як Python працює з HTTP-сервером?
- 53. Що відбувається, коли створюється віртуальне середовище?

### **Бази даних:**

- 54. Які є базові методи роботи з SQL- базою даних у Python?
- 55. Що таке SQL-транзакція?
- 56. Як зробити вибірку із SQL-бази з простою агрегацією?
- 57. Який вигляд має запит, який виконує JOIN між таблицями й до самих себе?
- 58. Як відправляти запити у SQL-базу даних без ORM?

**Алгоритми:**

59.Що таке алгоритми (наприклад, Big-O notation)?

60.Які є базові алгоритми сортування?

61.Що таке Bubble Sort і як це працює?

62.Що таке лінійна складність сортування?