|  |  |
| --- | --- |
| 1. Які є типи даних у PHP? | Числові типи - int, float  Строкові - string  Булеві - true, false  Масив та асоціативний масив  обʼект класу  Null |
| 2. Що таке асоціативний масив? | Це масив в якому данні зберігаються парами, перший в парі то ключ, а друге то значення |
| 3. Що таке константа у PHP і як її оголосити? | це змінна, значення якої не можна змінити після оголошення. оголошується через const або define() |
| 4. Що таке isset() і empty()?  5. Що таке суперглобальні змінні у PHP? | isset() - Перевіряє, чи змінна існує і не дорівнює NULL.  empty() - Перевіряє, чи змінна “порожня”  це вбудовані масиви, які завжди доступні в будь-якому місці коду без потреби передавати їх через global.  $GLOBALS - Асоціативний масив з усіма глобальними змінними, доступний з будь-якого місця.  $\_COOKIE - Дані з кукі, надіслані клієнтом.  $\_SESSION - Змінні поточної сесії користувача. |
| 6. Різниця між include, require, include\_once, require\_once.  7. Що таке інтерполяція рядків у PHP? | це функції для підключення файлів  include — використовується, коли файл не критичний для роботи скрипта. Якщо файл не знайдено, буде помилка, але скрипт продовжить виконання.  require — використовують, коли файл обов’язковий. Якщо файл відсутній, виконання скрипта припиняється  приставка once до цих функцій гарантує, що файл буде підключений тільки один раз  це вставка значень змінних безпосередньо всередину рядка,  $name = "Стас";  "Привіт, $name!"; |
| 8. Як у PHP видалити елемент масиву? | unset() - видаляє по ключу або по індексу, але індекси не змінюються  array\_filter() - фільтрування масиву за умовою |
| 9. Що таке передача параметрів за посиланням? | якщо ми хочемо змінити змінну поза функцією, то ми передаємо змінну через амперсанд(&)  &$num |
| 10. Що таке рекурсія? | це коли функція викликає сама себе  головна умова щоб вона закінчилась це мати умову для виходу з неЇ |
| 11. Як перевірити, чи існує функція? | за допомогою вбудованої функції **function\_exists()** |
| 12. Що таке анонімні функції (Closures)? | це функція без імені, яку можна присвоїти змінній, передати як аргумент іншій функції або повернути з функції. |
| 13. Як передати змінну у анонімну функцію через use? | через use ми передаємо змінну з зовнішньої області видимості  $func = function($x) use ($y) {  return $x \* $y;  }; |
| 14. Що таке клас і об’єкт? | Клас — це шаблон для створення об’єктів, де описані атрибути) та методи  Об’єкт — це змінна, створена за цим шаблоном |
| 15. Що таке властивості та методи класу? | Властивості вони ж атрибути — це змінні всередині класу, які описують стан об’єкта (наприклад, колір, розмір).  Методи — це функції всередині класу, які описують поведінку об’єкта (наприклад, рухатися, говорити). |
| 16. Що таке інкапсуляція? | **приховування внутрішніх деталей реалізації класу** від зовнішнього світу, щоб захистити дані і забезпечити контрольований доступ. public — доступ звідусіль   * protected — доступ лише всередині класу та його наслідників * private — доступ тільки всередині самого класу |
| 17. Що таке наслідування? | Це механізм, коли один клас (похідний) успадковує властивості і методи іншого класу (батьківського), щоб повторно використовувати код і розширювати функціонал. Extends() |
| 18. Що таке поліморфізм? | Це здатність об’єктів різних класів реагувати на один і той самий метод по-різному. |
| 19. Що таке інтерфейс? | шаблон, який визначає, які методи обов’язково має реалізувати клас  Сам інтерфейс не містить реалізації методів — тільки їхню назву, параметри. |
| 20. Що таке абстрактний клас? | – это класс, который нельзя создать напрямую (нельзя сделать new), он служит **шаблоном** для других классов. Абстрактный класс удобно использовать, когда нужно задать **общий функционал + обязательные методы** для всех наследников. |
| 21. Що таке трейт (trait)? | так як PHP не підтримує множинне наслідування, це механізм для повторного використання коду в різних класах, без необхідності наслідування підключається через use |
| 22. Що таке простори імен (namespaces)? | це спосіб групувати класи, функції та константи в окремі області, щоб уникнути конфліктів імен між різними частинами коду або бібліотеками. |
| 23. Що таке конструктор і деструктор? | Конструктор - метод, який автоматично викликається при створенні об’єкта класу. Використовується для ініціалізації атрибутів.  Деструктор - метод, який викликається, коли об’єкт знищується, використовується для якихось дій при завершенні. |
| 24. Що таке статичні методи і властивості? | належать самому класу, а не конкретному об’єкту. Їх можна викликати без створення екземпляру класу. |
| 25. Як у PHP заборонити створення екземпляру класу? | -Оголосити конструктор як private або protected  -Оголосити клас як abstract |
| 26. Що таке final клас або метод? | final клас — це клас, від якого не можна успадковувати  final метод — це метод, який не можна змінити в kласах-нащадках. |
| 27. Що таке this і self? | $this — посилання на поточний об’єкт, використовується всередині методів для доступу до властивостей і методів об’єкта.  $self — посилання на поточний клас, використовується для звернення до статичних властивостей та методів класу. |
| 28. Як реалізувати методи-ланцюжки (method chaining)? | це стиль виклику, коли один метод повертає об’єкт, що дозволяє послідовно викликати інші методи в одному виразі.  $builder->addHello()->addWorld()->getText(); |
| 29. Як працює \_\_toString()? | це **магічний метод**, який викликається, коли об’єкт намагаються використати як рядок |
| 30. Що таке магічні методи (\_\_get, \_\_set, \_\_call)? | \_\_get($name) - Викликається при спробі звернутися до неіснуючого або недоступного атрибута класу.  \_\_set($name, $value) - Викликається при спробі записати значення в неіснуючий або недоступний атрибут.  \_\_call($name, $arguments) - Викликається при спробі викликати неіснуючий або недоступний метод. |
| 31. Чим відрізняється json\_encode() від serialize()? | json\_encode() - для передачі PHP даних в форматі JSON  serialize() - перетворює PHP-дані у стрічку з форматом, зрозумілим PHP, включаючи всі типи |
| 32. Що таке регулярні вирази в PHP? | це інструмент для пошуку, перевірки, заміни і витягання частин тексту за певними шаблонами.  preg\_match() — шукає збіг за шаблоном;  preg\_match\_all() — шукає всі збіги;  preg\_replace() — замінює текст за шаблоном; |
| 33. Як перевірити формат email у PHP? | filter\_var($email, FILTER\_VALIDATE\_EMAIL);  FILTER\_VALIDATE\_EMAIL - Це константа фільтра в PHP, яка використовується з функцією filter\_var() для перевірки, чи є рядок валідною email-адресою. |
| 34. Як у PHP працює оператор @? | це оператор який приховує вивід помилки |
| 35. Як об’єднати два масиви? | за допомогою +, або функції array\_merge() |
| 36. Чим відрізняється array\_merge() від + для масивів? | + - не замінює значення з першого массиву на значення з другого, а array\_merge() замінює |
| 37. Що робить array\_map()? | застосовує передану функцію **до кожного елемента масиву** і повертає **новий масив** з результатами. |
| 38. Що таке array\_filter()? | Фільтрує елементи масиву через callback-функцію, повертає новий масив. |
| 39. Як порахувати кількість елементів у масиві? | Використати count($array). |
| 40. Як знайти елемент у масиві? | |  | | --- | |  |  |  | | --- | | in\_array($value, $array) або array\_search($value, $array) (повертає ключ). | |  | |  | |
| 41. Що таке array\_key\_exists()? | Перевіряє, чи існує заданий ключ у масиві. |
| 42. Як отримати ключі масиву? | |  | | --- | | Використати array\_keys($array). | |  |  |  | | --- | |  | |
| 43. Як відсортувати масив за ключами? | ksort($array) – за зростанням ключів, krsort($array) – за спаданням. |
| 44. Як знайти довжину рядка? | Використати strlen($string). |
| 45. Що робить substr()? | Повертає підрядок з вказаної позиції і довжини. substr("Hello", 1, 3); // ell |
| 46. Що таке str\_replace()? | Замінює всі входження підрядка іншим рядком. str\_replace("cat", "dog", "cat runs"); // dog runs |
| 47. Як змінити регістр символів у рядку?  48. Що таке trim()? | strtolower() - Перетворює всі символи рядка у нижній регістр  strtoupper() - Перетворює всі символи рядка у верхній регістр  ucfirst() - Перетворює перший символ рядка у велику літеру, решта залишається без змін.  ucwords() - Перетворює першу літеру кожного слова у велику.  Видаляє пробіли та вказані символи з початку і кінця рядка.  trim(" hi "); // "hi" |
| 49. Чим відрізняється htmlspecialchars() від strip\_tags()? | htmlspecialchars() - робить так щоб HTML-символи відображалися як текст, а не інтерпретувалися браузером.  strip\_tags() - Видаляє HTML і PHP-теги з рядка, залишаючи тільки текст. |
| 50. Що таке \_\_clone()? | Викликається при клонуванні об’єкта. |
| 51. Чим відрізняється self:: від static::? | Ключові слова self:: та static:: - використовуються для звернення до методів або властивостей класу без створення екземпляра  self - Посилається строго на клас, у якому написаний код.  static - Враховує клас, з якого викликано метод. |
| 52. Як реалізувати Singleton? | 1. Ініціалізуємо приватний статичний атрибут $instance  2.Ініціалізуємо приватний конструктор  3. РОбимо методи \_\_clone і \_\_wakeup приватними  4. Ініціалізуємо публічний статичний метод getInstance(), який повертає єдиний інстанс, створюючи його один раз |
| 53. Що таке Dependency Injection? | це коли клас не створює самі обʼєкти через new(), а отримує їх ззовні через конструктор тощо |
| 54. Що таке SOLID-принципи? | це набір п’яти принципів об’єктно-орієнтованого програмування, які допомагають писати **чистий, підтримуваний та гнучкий код**. |
| 55. Що таке патерн Factory? | це патерн проєктування, який **створює об’єкти без прямого використання оператора new** у коді, який використовує ці об’єкти. |
| 56. Що таке cookie? | це дані користувача які веб-сервер зберігає у браузері, для збереження інформації між запитами. |
| 57. Що таке session у PHP? | це механізм збереження **даних користувача на сервері** між HTTP-запитами |
| 58. Що таке CSRF і як від нього захиститися? | це атака, при якій зловмисник змушує користувача виконати небажану дію на сайті, на якому він вже авторизований.  захиститись можна за допомогою Використання CSRF-токенів |
| 59. Що таке XSS і як його уникнути? | це вразливість, яка дозволяє зловмиснику вставити свій JavaScript-код у вебсторінку, щоб він виконався у браузері жертви. |
| 60. Як працюють індекси в SQL | Індекс у базі даних — це **спеціальна структура даних** (зазвичай B-Tree або Hash), яка прискорює пошук рядків у таблиці за певними колонками.  Він працює як **зміст у книзі**: щоб знайти сторінку з потрібним словом, ти дивишся в зміст, а не гортаєш усю книгу. Їх варто створювати для колонок, які часто використовуються у WHERE, JOIN, ORDER BY, але не треба індексувати все підряд. |
| 61. как создать индекс?  62. как порядок в индексе влияет на производительность? | CREATE INDEX idx\_users\_email ON users(email);  CREATE UNIQUE INDEX idx\_users\_email\_unique ON users(email); - Забезпечує унікальність значень у колонці  Видалення - DROP INDEX idx\_users\_email ON users;  Перевірка використання індексу - EXPLAIN SELECT \* FROM users WHERE email = '[test@example.com](mailto:test@example.com)';  Дивись, чи у плані запиту вказано Index Scan або Using index.   1. «Индекс — это как отсортированный список. Порядок колонок в индексе важен, потому что база ищет слева направо.  Например, если индекс по (фамилия, имя), то: 2. запрос по фамилии будет быстрым, 3. запрос по фамилии и имени вместе тоже быстрый, 4. а вот только по имени — индекс не поможет, потому что он идет после фамилии. 5. Поэтому порядок колонок в индексе нужно подбирать в зависимости от того, какие запросы чаще всего выполняются». |
| 63. Що таке REST API? | це стиль архітектури для веб-сервісів, яким клієнт може спілкуватися з сервером через інтернет.  Клієнт запитує ці ресурси через HTTP-запити:  GET – отримати дані  POST – створити нові дані  PUT/PATCH – змінити дані  DELETE – видалити дані |
| 64. Що таке CRUD? | це базова концепція роботи з даними в будь-якому додатку. |
| 65. git fetch  66. git pull | |  | | --- | |  |  |  | | --- | | Завантажує оновлення з віддаленого репозиторію, **не зливає** їх з локальною гілкою | | Завантажує оновлення з віддаленого репозиторію і **автоматично зливає** їх з локальною гілкою (fetch + merge | |
| 67. git rebase | Переносить локальні коміти на іншу базу. Она используется, чтобы **переписать историю коммитов**. |
| 68. git log | Показує історію комітів |
| 69. git diff | Показує різницю між файлами/комітами |
| 70. git squash | Об’єднує кілька комітів в один (зазвичай через rebase) |
| 71. git cherry-pick | |  | | --- | | Застосовує конкретний коміт з іншої гілки у поточну |  |  | | --- | |  |   Коли потрібно взяти один коміт без злиття всієї гілки |
| 72 git add | помещает изменения в staging area, чтобы указать Git, что именно должно попасть в следующий коммит». |
| 73. git reset | предназначена для отмены коммитов и изменений, перемещая указатель HEAD на предыдущий коммит. Меняя при этом историю |
| 74. **git revert** | создаёт **новый коммит**, который отменяет выбранный коммит по ID. История сохраняется |
| 75. git clone | создаёт **локальную копию** удалённого репозитория на твоём компьютере. |
| 76. Git fork | создаёт **свою копию репозитория в облаке**, чтобы вносить изменения и потом предлагать их в оригинальный репозиторий через pull request. |
| 77 git restore | для отмены локальных изменений, возвращая файлы к состоянию последнего коммита |
| 78 git checkout | <ветка> — переключиться на ветку. -b <ветка> — создать и сразу переключиться на новую ветку. |