

Oop

-1 What is OOP

عبارة عن طريقه برمجته تم استخدامها لتجزئته البرنامج لأكثر من object ، وبشكل اخر يطلق عليها برمجته تجزيئيه لانها تأخذ كل جزء من البرنامج ويتم التعامل معه كوحده واحده ، ومن فوائد تطبيق ال oop كان تقليل حجم الكود المكتوب وبالتالي تسريع عمل البرنامج وتفاذي الاخطاء واكتشافها بطريقه اسهل ، اصبح الكود اكثر تنظيما

-2 ايه هي مبادئ ال OOP

Encapsulation, polymorphism, abstract, inheritance

Encapsulation:

Definition: Encapsulation is the bundling of data (attributes) and methods (behaviors) that operate on the data into a single unit, typically a class. •

Purpose: Encapsulation aims to hide the internal state of an object and restrict access to its data, providing controlled and secure access through well-defined interfaces (methods). •

Key Aspects:

 •

Data Hiding: Encapsulation hides the internal state of an object by making its data members private or protected, preventing direct access from external code. •

Access Control: Access to the object's data is provided through accessor (getter) and mutator (setter) methods, which enforce encapsulation and protect the integrity of the object's state. •

Information Hiding: Encapsulation encapsulates the implementation details of a class, allowing the class to change its internal representation without affecting external code. •

Example: A class representing a bank account may encapsulate its balance as a private field and provide methods like **Deposit** and **Withdraw** to manipulate the balance, ensuring that the balance is accessed and modified in a controlled manner. •

Abstraction:

Definition: Abstraction is the process of simplifying complex systems by modelling them at a high level of abstraction, focusing on essential details while hiding unnecessary complexity. •

Purpose: Abstraction aims to focus on the essential characteristics of an object or system while suppressing irrelevant details, providing a simplified and conceptual view of the system. •

Key Aspects:

 •

Generalization: Abstraction involves identifying common patterns and generalizing them into abstract concepts, classes, or interfaces that capture the essential properties and behaviors. •

Modeling Real-World Entities: Abstraction models real-world entities as abstract classes, interfaces, or concepts, focusing on what an object does rather than how it does it. •

Information Hiding: Abstraction hides unnecessary details and implementation complexities, allowing developers to focus on the high-level design and functionality of the system. •

Example: An abstract class **Shape** may define abstract methods like **Calculate Area** and **Draw**, representing common behaviors of geometric shapes without specifying how each shape calculates its area or draws itself. •

Inheritance:

Definition: Inheritance is a mechanism by which a class (subclass or derived class) can inherit the properties (fields and methods) of another class (superclass or base class). The subclass can then extend or modify the behavior of the superclass by adding new features or overriding existing ones. •

Purpose: Inheritance facilitates code reuse and promotes hierarchical organization by allowing classes to inherit common characteristics from a base class while specializing or customizing behavior in derived classes. •

Key Aspects:

Parent-Child Relationship: Inheritance establishes an "is-a" relationship between classes, where a subclass "is a" specialized version of its superclass. •

Code Reuse: Inheritance allows subclasses to reuse code and behavior defined in the superclass, reducing duplication and promoting modularity. •

Method Overriding: Subclasses can override methods inherited from the superclass to provide specialized implementations, enabling polymorphic behavior. •

Example: A **Vehicle** class may serve as a base class, with subclasses such as **Car**, **Truck**, and **Motorcycle** inheriting common properties and methods like **startEngine()** and **stopEngine()**. •

Polymorphism:

Definition: Polymorphism is the ability of objects of different types to respond to the same message or method invocation in different ways. It allows objects to exhibit different behaviors based on their actual types at runtime. •

Purpose: Polymorphism enables flexibility, extensibility, and dynamic behavior in object-oriented systems by allowing objects to be treated uniformly and manipulated at a higher level of abstraction. •

Key Aspects:

Method Overriding: Polymorphism is often achieved through method overriding, where a method in a subclass provides a specialized implementation that overrides the method in the superclass. •

Dynamic Binding: Polymorphic behavior is determined at runtime based on the actual type of the object being referenced, rather than the declared type of the reference variable. •

Subtype Polymorphism: Subclasses can be treated as instances of their superclass, allowing for substitution of derived types wherever the base type is expected. •

Example: Given a **Shape** superclass with a method **draw()**, polymorphism allows instances of subclasses like **Circle**, **Rectangle**, and **Triangle** to respond to **draw()** in their own specialized ways. •

3- ليه بنطبق مبدأ encapsulation مع اننا يمكننا استخدام getter & setters

تقييد الوصول والتعديل
خفاء التفاصيل التنفيذية
سهولة الصيانة والتعديل

4- ايه هوا ال constructors و كم cons نقدر ننشأ وازاي افرق بينهم

هو نوع خاص من الأساليب يتم استدعاؤه تلقائيًا عند إنشاء مثيل (كائن) لفئة ما. الغرض الأساسي منه هو تهيئة الكائن الذي تم إنشاؤه حديثًا، وإعداد حالته الأولية وتنفيذ أي عمليات إعداد ضرورية. ويمكن إنشاء أكثر من واحد و يمكننا التفريق بينهم عن طريق ال signature و ال parameters

5- ايه هوا ال overriding

يسمح لفئة فرعية (أو فئة مشتقة) بتطبيق محدد لطريقة تم تعريفها بالفعل في الفئة الأساسية.

6- ايه هيا ال access modifiers مع شرح كل واحد منهم

يسمح بالوصول إلى member من أي code آخر في نفس class أو من class الأخرى التي تشير إليه.	Public
يقيّد الوصول إلى العضو فقط داخل نفس class أو البنية التي تم الإعلان عنها. لا يمكن الوصول إلى private member.	Private
يسمح بالوصول إلى العضو ضمن فئته أو بنيته ومن خلال child class. يتم استخدامه بشكل شائع لتزويد child class بإمكانية الوصول إلى أعضاء معينين مع تقييد الوصول إلى class الأخرى.	protected
يقيّد الوصول إلى الأعضاء داخل نفس الكلاس بحيث لا يمكن الوصول إلى الأعضاء المشار إليهم ب internal إلا من خلال نفس الكلاس	internal
يسمح بالوصول إلى العضو من خلال التجميع الذي يحتوي عليه ومن خلال الفئات المشتقة، بغض النظر عما إذا كانوا في نفس التجميع أو في تجميع مختلف. فهو يجمع بين إمكانية الوصول إلى كل من المحمية والداخلية.	protected internal

7- Difference between constructors and destructors

Constructor	destructor	
طريقة خاصة تستخدم لتهيئة object من class. يتم استدعاؤه تلقائيًا عند إنشاء instance من class.	يستخدم عند تدمير كائن أو خروجه عن النطاق.	purpose
يتم استدعاء constructor تلقائيًا عند إنشاء instance للـ class باستخدام الكلمة الأساسية الجديدة. وهي تضمن تهيئة object بشكل صحيح قبل استخدامه.	يتم استدعاؤه تلقائيًا عند تدمير كائن أو خروجه عن النطاق.	Invocation
المنشئون لديهم نفس اسم الفئة وليس لديهم نوع إرجاع. قد تحتوي على parameters، مما يسمح بطرق مختلفة لتهيئة الكائنات.	يكون للمدمرات نفس اسم الفئة التي يسبقها رمز التلدة (~). إنهم لا يأخذون أي parameters وليس لديهم نوع إرجاع.	syntax
يتم استخدام المنشئين لتهيئة حالة الكائن، مثل تهيئة الحقول، أو تعيين قيم الخصائص الافتراضية، أو تنفيذ مهام التهيئة الأخرى.	يتم استخدام Destructors لتحرير الموارد التي يحتفظ بها كائن، مثل إغلاق الملفات، أو تحرير الذاكرة، أو تنفيذ مهام التنظيف الأخرى.	usage

8- عرف ال abstract

يتم استخدامها لتحديد class المجردة والـ members المجردين داخل تلك classes. لا يمكن إنشاء فئة مجردة مباشرة؛ بدلاً من ذلك، فهو بمثابة مخطط لكي ترث منه الفئات الأخرى. يتم الإعلان عن الأعضاء المجردين ضمن الفئات المجردة دون توفير تطبيق ويجب تنفيذهم بواسطة فئات مشتقة غير مجردة.

9- What is copy constructor

يقوم بتهيئة كائن جديد كنسخة من كائن موجود. يقوم بإنشاء مثيل جديد لفئة لها نفس الحالة مثل مثيل آخر.

10- Difference between abstract and interface

Abstract	Interface	Diff
يتم استخدام فئة مجردة عندما تريد تحديد فئة أساسية لا يمكن إنشاء مثيل لها مباشرة وقد تحتوي على أعضاء مجردة وغير مجردة (ملموسة).	يتم استخدام الواجهة لتحديد العقد، وتحديد مجموعة من الأساليب أو الخصائص أو الأحداث أو المفهرسات التي يجب أن توفرها فئات التنفيذ.	Purpose
توفر فئة مجردة تفاصيل التنفيذ ويمكن أن تحتوي على حقول ومنشآت وأساليب مع عمليات التنفيذ. وهو يدعم الوراثة المفردة، مما يعني أن الفصل يمكن أن يرث من فصل مجرد واحد فقط.	تدعم الواجهة الوراثة المتعددة، مما يعني أن الفصل يمكنه تنفيذ واجهات متعددة. ومع ذلك، لا يمكن أن تحتوي الواجهات على تفاصيل التنفيذ؛	Inheritance
يتم استخدام الفئات المجردة عندما تريد توفير تطبيق أساسي مشترك للفئات المشتقة مع السماح أيضًا بمرونة التخصيص من خلال الأساليب التي تم تجاوزها. إنها مفيدة لإعادة استخدام التعليمات البرمجية وإنشاء التسلسلات الهرمية للفصل.	يتم استخدام الواجهات عندما تريد تنفيذ عقد دون توفير أي تنفيذ. إنها مفيدة لتحقيق تعدد الأشكال والسماح بمعاملة الفئات المختلفة بشكل موحد بناءً على سلوكهم المشترك.	Usage
يمكن أن تحتوي الفئات المجردة على معدلات وصول مثل عامة أو محمية أو داخلية مطبقة على أعضائها، مما يسمح لك بالتحكم في الرؤية وإمكانية الوصول.	يمكن أن تحتوي الفئات المجردة على معدلات وصول مثل عامة أو محمية أو داخلية مطبقة على أعضائها، مما يسمح لك بالتحكم في الرؤية وإمكانية الوصول.	Access Modifiers
يمكن أن تحتوي الفئات المجردة على أعضاء مجردين (بدون تطبيق) وغير مجردين (مع التنفيذ).	يمكن أن تحتوي الواجهات فقط على إعلانات الطريقة أو الخاصة أو الحدث أو المفهرس. لا تحتوي على أي تطبيق ولا يمكن أن تحتوي على حقول أو منشآت أو طرق ملموسة.	Member Definitions

11- ازاي اقدر اطبق ال encapsulation (مثال عليها في ال OOP)

التغليف في البرمجة الشيئية (OOP) هو مفهوم تجميع data (السمات أو الخصائص) و methods (السلوك أو الوظائف) التي تعمل على البيانات في unit أو class واحدة. فهو يسمح لك بالتحكم في الوصول إلى الحالة الداخلية للكائن، ومنع التلاعب المباشر من خارج الفصل وفرض تكامل البيانات. يساعد التغليف في تحقيق إخفاء البيانات وتجريدها ونمطيتها في التعليمات البرمجية الخاصة بك.

```

public class Person
{
    // Private fields
    private string name;
    private int age;

    // Public properties with getter and setter methods
    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    public int Age
    {
        get { return age; }
        set
        {
            if (value >= 0 && value <= 120) // Validate the age
            {
                age = value;
            }
            else
            {
                throw new ArgumentOutOfRangeException("Age must be between 0 and 120.");
            }
        }
    }

    // Constructor
    public Person(string name, int age)
    {
        Name = name; // Use property setter
        Age = age;    // Use property setter
    }

    // Public method
    public void DisplayInfo()
    {
        Console.WriteLine($"Name: {Name}, Age: {Age}");
    }
}

```

virtual function and pure virtual function -12

virtual function

pure virtual function

<ul style="list-style-type: none"> ○ is a method declared in a base class with the virtual keyword. • When a method is marked as virtual, it means that it can be overridden by derived classes. 	<ul style="list-style-type: none"> • C# does not have a direct equivalent of a pure virtual function as in C++. • However, you can achieve a similar concept using abstract methods in abstract classes or interfaces. • An abstract method is a method declared in an abstract class or interface without providing an implementation. It is terminated with a semicolon (;). • Abstract methods serve as placeholders for methods that must be implemented by derived classes. • Abstract classes cannot be instantiated directly and may contain one or more abstract methods that must be overridden by derived classes. • Interfaces define a contract without providing any implementation, and all methods in an interface are implicitly abstract.
--	--

Can constructor be overridden -13

yes

What is setter and getter -14

الحروف والمحددات من الأساليب المستخدمة للوصول إلى الحقول الخاصة (الخصائص) للفئة وتعديلها. إنها توفر وصولاً متحكمًا إلى سمات الفصل، مما يسمح لك بفرض التغليف والحفاظ على سلامة بيانات الفصل.

Get	Set
<p>يتم استخدام طريقة getter (المعروفة أيضًا باسم طريقة الوصول) لاسترداد القيمة الحالية للحقل الخاص.</p> <p>وعادةً ما تحتوي على بادئة get متبوعة باسم الخاصية.</p> <p>تُستخدم الحروف عادةً لتوفير وصول للقراءة فقط إلى الحقول الخاصة، مما يسمح للتعليمات البرمجية الخارجية باسترداد القيمة دون الوصول مباشرة إلى الحقل.</p>	<p>يتم استخدام طريقة setter (تُعرف أيضًا باسم طريقة mutator) لتعديل قيمة الحقل الخاص.</p> <p>تحتوي عادةً على بادئة محددة متبوعة باسم الخاصية، وتأخذ معلمة تمثل القيمة الجديدة.</p> <p>يتم استخدام أدوات الضبط بشكل شائع لتوفير الوصول للكتابة فقط أو القراءة والكتابة إلى الحقول الخاصة، مما يسمح للتعليمات البرمجية الخارجية بتعديل القيمة أثناء فرض التحقق من الصحة أو منطق الأعمال الآخر.</p>

What is static -15

يستخدم للإعلان عن الأعضاء (الحقول والأساليب والخصائص والأحداث) التي تنتمي إلى النوع نفسه وليس إلى مثيلات النوع. يرتبط الأعضاء الثابتون بالنوع نفسه وليس بأي مثيل محدد من النوع.

Difference between reference type & value type -16

Aspect	Value Types	Reference Types
Memory location	Stack (or within objects)	Heap
Storage	Stores data directly	Stores a reference to data
Instances	Primitive types, structs	Classes, interfaces, arrays, strings
Assignment	Copies the data	Copies the reference
Size	Fixed size	Dynamic size
Copy behavior	Independent copies (pass by value)	Shared reference (pass by reference)
Examples	int, double, bool, struct	class, interface, array, string

What is abstract class -17

a class that cannot be instantiated directly and may contain one or more abstract members (methods, properties, indexers, and events) that do not have an implementation

What is abstract method -18

is a method declared in an abstract class or interface that does not contain an implementation.

Difference between overriding & overloading -19

overloading	overriding
<ul style="list-style-type: none">Overloading occurs when a class has multiple methods with the same name but different signatures (different parameters or parameter types).Overloaded methods are distinguished by their method signatures, which include the method name and the types and/or number of parameters.Overloading allows methods with the same name to perform different tasks based on the arguments passed to them.Overloading is resolved at compile time based on the number and types of arguments provided in the method call.	<ul style="list-style-type: none">Overriding occurs when a derived class provides a specific implementation for a method that is already defined in its base class.Overriding allows a subclass to provide its own implementation of a method inherited from its base class, providing a way to customize or extend the behaviour of the base class.Overriding is used in inheritance scenarios, where a subclass inherits behaviour from its superclass but needs to customize that behaviour.

20- هل ينفع ال constructor يكون private

ايوة ينفع ، ولما هستخدم private constructor همنع ان يتعمل object من ال class بتاعتي اعتمادا على نوع ال constructor اللي اتعلمه private

21- هل ال constructor له return type

لا

22- ايه هوا التيكنيك اللي اقدر اربط بيها ال classes ببعضها

فيه 3 اجابات صح للسؤال ده

1- هتقول فيه للربط بين ال classes بستخدم concept of inheritance

2- لو قالك حاجه غير الوراثه هيبقى قدامك اجابتين

• هتقولو في طريقه لتوضيح الترابط بين تنفيذ 2 classes مع بعض

وارتباطهم ببعض بيكون ب 3 طرق

➤ Association : وده الطريقة العامه ان فيه 2 classes مرتبطتين ببعض ف أحد

ال classes دي واخده ميثود او فانكشن او حتى متغير من الكلاس التانيه

ومعتمده بشكل عام على الكلاس دي في شغلها

➤ Aggregation : ودي بتعتبر weak type of association وبيتم وصف

الترابط بين 2 classes انه aggregation لما الكلاس a اللي واخد ميثود او

اي information من الكلاس التانيه b ، ولكن مع انتهاء او موت الكلاس b

وهو الكلاس المأخوذ منه المعلومات ، يمكن للكلاس a ان يتابع عمله دون

الحاجه لوجود الكلاس b

➤ Composition : وهي عبارته عن strong type of association وبيتم

وصف العلاقه بين اثنين كلاس بكدا لما بيكون عمل احد الكلاس معتمد بشكل

اساسي على الكلاس الاخرى بحيث لو واحده انتهت او تم حذفها لا يمكن للاخرى

العمل وتتوقف هي الاخرى

- وفيه طريقه تانيه وهو استخدام مصطلح ال partial class وهو عبارته عن تقسيم ال class الي اكثر من جزء كل جزء منها في file ، حيث اكبر فائدة لها هي انه يمكن لكل programmer ان يعمل على الجزء المخصص له من ال class بشكل منفرد في نفس الوقت

23- ينفع استدعي instance member جوا static method مع الشرح

Static method لا يمكن ان تحتوي على اي عنصر الا ويجب ان يكون static

لذا لا يمكن استدعاء instance member موجود بداخل static method بشكل مباشر كما نستدعي

ال static member التي بداخل ال static method

24- ايه الفرق بين ال semantic error and logical error وايهم يعطل عمل الكود

- شكل مبسط ال logical error عبارة عن خطأ رياضي او خطأ يمكن ان يسبب تغيير ال output المتوقع وحدوث اخطاء في الجورزم المسأله ؛ هو لا يسبب توقف لتنفيذ البرنامج لكن يسبب اخطاء في ال output النهائي مما يحدث اخطاء جسيمه
- Semantic errors : عندما لا يظهر خطأ في ال logic او خطأ في ال syntax فغالبا ما يكون خطأ semantic بمعنى ان يكون خطأ في المعنى البرمجي او مثلا خطأ في تحديد وحدات القياس للمسائل وهكذا ؛ فهو خطأ في المعنى

Logical Error	semantic error
<ul style="list-style-type: none">• A logical error occurs when the code does not produce the expected output due to a flaw or mistake in the algorithm or program logic.• Logical errors are typically caused by incorrect assumptions or errors in the design or implementation of the algorithm, rather than syntax or language usage.• These errors often result from errors in the flow of control, incorrect conditional logic, or incorrect calculations.• Logical errors can be more challenging to detect and debug than syntax errors because the code may compile and execute without errors, but the output may not be what was expected.	<ul style="list-style-type: none">• A semantic error occurs when the code does not behave as intended due to incorrect usage of programming language constructs or misinterpretation of specifications.• Semantic errors are often related to the meaning or semantics of the code rather than its syntax.• These errors may result from incorrect assumptions about how a particular language feature or function works, or from misunderstanding the requirements of the problem being solved.

25- What is sealed method

the **sealed** modifier is used to prevent a method from being overridden in derived classes. When a method is marked as **sealed**, it means that it cannot be further overridden in any subclass.

26- ينفع يبقى فيه static class ولية

a static class is a special type of class that cannot be instantiated and can only contain static members (fields, methods, properties, and events). Static classes are commonly used to group related utility methods or constants together in a single unit, providing a convenient way to organize and access shared functionality without the need to create instances of the class.

27- يعني ايه مصطلح this

keyword is a reference to the current instance of the class in which it appears. It is primarily used within instance methods and instance property accessors to refer to the current object on which the method or property is being invoked.

Difference between method overriding & method hiding -1

Method Overriding:	Method Hiding:
<ul style="list-style-type: none">• Method overriding occurs when a method in a derived class provides a new implementation for a method that is already defined in its base class.• The method in the derived class must have the same signature (name, return type, and parameters) as the method in the base class.• Method overriding is achieved by using the override keyword in the derived class.• When a method is overridden, the runtime polymorphism mechanism ensures that the correct method implementation is called based on the actual type of the object.• Method overriding is used to extend or modify the behavior of the base class method in the derived class, allowing for dynamic method dispatch.	<ul style="list-style-type: none">• Method hiding (also known as method shadowing) occurs when a method in a derived class declares a method with the same name as a method in its base class.• Unlike method overriding, method hiding does not require the method in the derived class to have the same signature as the method in the base class.• Method hiding is achieved by using the new keyword in the derived class.• When a method is hidden, the method in the derived class completely replaces the method with the same name in the base class for instances of the derived class.• Method hiding does not participate in runtime polymorphism; the method to be called is determined statically based on the declared type of the object.

Database

-2 Student & courses نوع العلاقة ما بينهم وكيفيه ربط الجدولين ببعض + with sql عبر عن (كل طالب مسجل كام كورس)

العلاقة بين الطلاب (Students) والدورات (Courses) هي علاقة "Many-to-Many"، حيث يمكن للعديد من الطلاب أن يكونوا مسجلين في العديد من الدورات والعكس صحيح أيضًا، وهذا يعني أن هناك جدول وسيط (Intermediate Table) يربط بين الطلاب والدورات.

-3 Write SQL بيجيب اعلى خمس مرتبات من جدول الموظفين

للحصول على أعلى خمس مرتبات من جدول الموظفين (Employees) في SQL، يمكنك استخدام عبارة الاستعلام التالية باستخدام العملية ORDER BY لترتيب النتائج بناءً على الراتب (salary) بترتيب تنازلي (DESC)، واستخدام العبارة LIMIT لاستعادة أعلى خمس سجلات.

-4 Difference between DML & DDL

DML (لغة معالجة البيانات) وDDL (لغة تعريف البيانات) كلاهما مجموعتان فرعيتان من SQL (لغة الاستعلام المنظمة) المستخدمة في أنظمة إدارة قواعد البيانات (DBMS) للتفاعل مع قواعد البيانات. ومع ذلك، فإنها تخدم أغراضًا مختلفة وتعمل على جوانب مختلفة لإدارة قاعدة البيانات. فيما يلي تفاصيل الاختلافات بين DML وDDL:

DML	DDL
<ul style="list-style-type: none">• Purpose: DML is used to manipulate (insert, update, delete, and retrieve) data stored in the database tables.• Operations:<ul style="list-style-type: none">• INSERT: Adds new records (rows) to a table.• UPDATE: Modifies existing records (rows) in a table.• DELETE: Removes existing records (rows) from a table.• SELECT: Retrieves data from one or more tables based on specified criteria.	<ul style="list-style-type: none">• Purpose: DDL is used to define the structure of the database schema, including tables, indexes, views, constraints, and other database objects.• Operations:<ul style="list-style-type: none">• CREATE: Creates new database objects such as tables, views, indexes, etc.• ALTER: Modifies the structure of existing database objects.• DROP: Deletes existing database objects from the schema.• Focus: DDL focuses on defining and managing the database schema and its components, such as tables,

Difference between truncate and delete and drop -5

بشكل مختصر

Drop -1

- يتمسح ال table كله من الداتا بيز وكمان يتمسح المكان المحجوز له في الميموري (يتمسحه مسح كلي)
- DDL commands
- اسرع من delete وأبطأ من ال truncate ؛ بتبدأ بمسح ال rows الاول وبعد كدا بنيه ال table
- DROP TABLE table_name

Delete -2

- يتسمح واحد او اكثر من record (rows) من ال table ، بس مش يتمسح المكان المحجوز للجدول في الميموري (مش بتأثر على بنية ال table)
- DML commands
- اقل سرعه من كل من ال drop & truncate وده لانها بتحذف row واحد خلال الوقت حسب شرط معينه
- DELETE FROM table_name WHERE conditions;

Truncate -3

- بتحذف كل ال rows من الجدول ، بس مش يتمسح المكان المحجوز للجدول في الميموري (مش بتأثر على بنية ال table)
- اسرع طريقه للحذف لان يتمسح كل ال records في نفس الوقت بدون اي شروط
- TRUNCATE TABLE table_name;

للمزيد من المعلومات انظر الجدول التالي



TRUNCATE	DELETE	DROP
<ul style="list-style-type: none"> • Purpose: TRUNCATE is used to remove all rows from a table quickly and efficiently. • Operation: TRUNCATE removes all rows from a table while preserving the table structure (columns, indexes, constraints, etc.). • Transaction: TRUNCATE is a DDL (Data Definition Language) operation and is typically not transactional. It cannot be rolled back within a transaction. • Performance: TRUNCATE is usually faster than DELETE, especially for large tables, as it does not generate individual delete operations for each row. It deallocates data pages instead of removing rows one by one. • Reset Identity Column: In some databases, TRUNCATE resets the identity column (auto-increment) to its initial seed value 	<ul style="list-style-type: none"> • Purpose: DELETE is used to remove specific rows from a table based on a specified condition or criteria. • Operation: DELETE removes rows from a table based on the specified WHERE clause, allowing selective deletion. • Transaction: DELETE is a DML (Data Manipulation Language) operation and is transactional. It can be rolled back within a transaction if necessary. • Performance: DELETE may be slower than TRUNCATE, especially for large tables, as it generates individual delete operations for each row. • Logging: DELETE operations are typically logged in the database transaction log, which can affect database performance and log file size. 	<ul style="list-style-type: none"> • Purpose: DROP is used to remove database objects such as tables, views, indexes, or stored procedures from the database. • Operation: DROP permanently removes the specified database object and all associated data or definitions. • Transaction: DROP is a DDL operation and is not transactional. Once a database object is dropped, it cannot be rolled back. • Irreversible: DROP is irreversible and permanently deletes the database object and all associated data or definitions. • Cascading: Some databases support cascading options with DROP, allowing you to automatically drop dependent objects (e.g., dropping a table drops associated indexes and constraints).

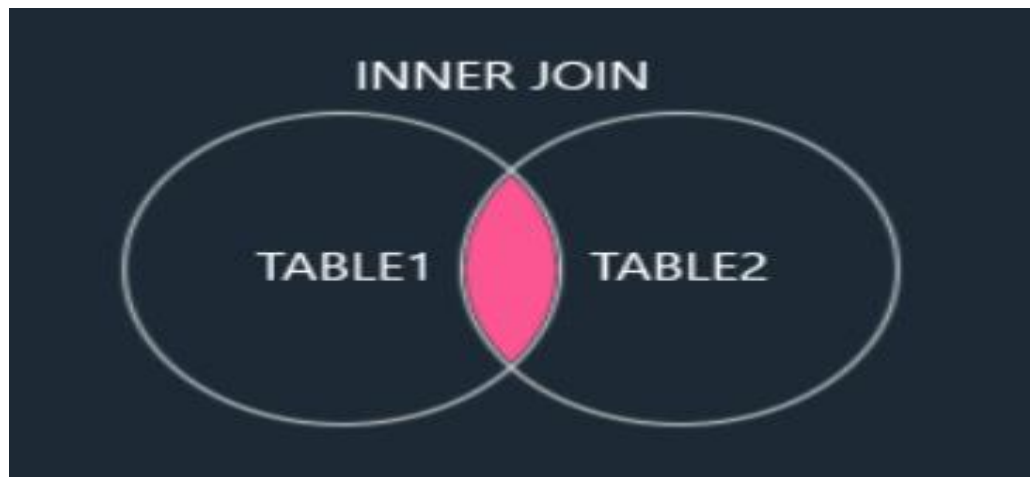
6- ال **inner join query**

.The INNER JOIN keyword selects records that have matching values in both tables

Example

Join Products and Categories with the INNER JOIN keyword:

```
SELECT ProductID, ProductName, CategoryName
FROM Products
INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID;
```



7- كود sql طلع كل الموظفين اللي اسمهم زياد

```
SELECT*
FROM Employees
WHERE employee_name= "زياد"
```

8- اضافته صف جديد بال sql

```
INSERT INTO Employees (employee_name, salary, department_id)
VALUES ('John', 5000, 1);
```

9- طريقه عمل ال insert with SQL

- Specify the Table ✓
- Specify Columns ✓
- Specify Values ✓
- Execute the Query ✓

10- What is query

هو أمر أو عبارة مكتوبة بلغة استعلام محددة (مثل SQL) يتم استخدامها للتفاعل مع نظام إدارة قاعدة البيانات (DBMS) لاسترداد البيانات المخزنة في قاعدة البيانات أو معالجتها أو تعديلها.

What is subquery -11

الاستعلام الفرعي، المعروف أيضًا باسم الاستعلام الداخلي أو الاستعلام المتداخل، هو استعلام SQL متداخل ضمن استعلام SQL آخر. يتم استخدامه لاسترداد البيانات أو إجراء العمليات الحسابية التي تعتمد على نتائج استعلام آخر. يتم وضع الاستعلامات الفرعية بين قوسين ويمكن استخدامها في أجزاء مختلفة من عبارة SQL، مثل عبارات SELECT أو INSERT أو UPDATE أو DELETE.

What is a relationship and what are they -12

تشير العلاقة إلى الارتباط أو الاتصال بين جدولين أو أكثر بناءً على الحقول أو الأعمدة المشتركة. تحدد العلاقات كيفية ارتباط البيانات الموجودة في جدول واحد بالبيانات الموجودة في جدول آخر. إنها أساسية لأنظمة إدارة قواعد البيانات العلائقية (RDBMS) وتستخدم للحفاظ على سلامة البيانات وفرض القيود وتمكين استرجاع البيانات ومعالجتها بكفاءة. ويمكن تقسيم أنواع العلاقات في ال DB الى 3 اقسام

1- **Degree of relation** : وتعتبر عن كام entity مشتركه في العلاقه الواحده ف هي تنقسم الى انواع

- Binary : بين two entity
- Unary / recursive : بتكون بين entity ونفسه
- Ternary : علاقه بتربط بين 3 entity

2- **Cardinality ratio** : ويتحدد على حسب ال system او نظام المؤسسه فتنقسم الى

- One to many
- Many to many
- One to one

3- **Relation participation** : بتعتبر عن اقل عدد من ال relation اللي ممكن تحصل لل

- entity الواحد وينعبر عنها ب
- May
 - must

What is constrains -13

عبارة عن شويه قواعد بتتحط للبيانات الموجوده في ال tables زي مثلا ال datatype ، وبستخدم القواعد دي عشان اضمن صحة وسلامه البيانات اللي عندي

What is not null -14

عبارة عن نوع من انواع ال constrains اللي اقدر احطها على البيانات الموجوده عندي في ال table والجمله دي عشان اتأكد ان الخلية دي مش فاضيه / تحتوي على داتا / not null

What is CHECK -15

عبارة عن constrains بقدر بيها احدد رينج الفاليو بتاعت الداتا اللي عندي

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

What is DEFAULT -16

بتستخدم لوضع قيم مبدئية للخلية وبيتم اضافتها للخلية تلقائيا اذا لم يتم تحديد قيم مخصصه للخلية

What is UNIQUE -17

بتستخدم لجعل كل قيمه من قيم العمود مميزة لا تتكرر اي خلية في خلية اخرى
The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table

What is primary key -18

عبارة عن قيمة بتكون unique وكذلك not null ، مبتتكررش في اي قيمه من قيم العمود وهيا
للي بيها بقدر اميز كل عمود عن الثاني

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields)

What is foreign key -19

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

NoSQL، الذي يرمز إلى "ليس فقط SQL"، هو مصطلح يستخدم لوصف قواعد البيانات غير العلائقية التي توفر آلية لتخزين واسترجاع البيانات التي تم تصميمها بوسائل أخرى غير العلاقات الجدولية المستخدمة في قواعد البيانات العلائقية. تم تصميم قواعد بيانات NoSQL للتعامل مع كميات كبيرة من أنواع البيانات غير المنظمة أو شبه المنظمة أو متعددة الأشكال وتوفر المرونة وقابلية التوسع التي قد تواجهها قواعد البيانات العلائقية التقليدية في سيناريوهات معينة.

-21 اذكر علاقه كمثال على many to many & one to one

المثال مختصر:

- 1- Many to many : مثال عليها مجموعه من الطلبة ومجموعة كورسات
يمكن لمجموعة من الطلبة اخذ مجموعه من الكورسات / بمعنى اخر الطالب الواحد يمكنه اخذ
مجموعه من الكورسات و الكورس الواحد يمكن ان يؤخذ من خلال اكثر من طالب
- 2- One to one : مثال عليها المواطنين/الموظفين وال id الخاص بكل مواطن
لكل مواطن يجب ان يكون هناك id واحد فقط ، وكل رقم id يجب ان لا يتكرر لأكثر من شخص
واحد

المثال مطول مشروح جدا 😞

Many-to-Many Relationship Example: In a university database, consider the relationship between students and courses. A student can enroll in multiple courses, and a course can have multiple students enrolled in it. This forms a many-to-many relationship between the "Students" table and the "Courses" table.

- Students Table:
 - student_id (Primary Key)
 - student_name
 - ...
- Courses Table:
 - course_id (Primary Key)
 - course_name
 - ...

To represent the many-to-many relationship, you would typically introduce a third table, often called an "Enrollments" or "StudentCourses" table, which serves as a junction table to link students with the courses they are enrolled in:

- Enrollments Table:
 - enrollment_id (Primary Key)
 - student_id (Foreign Key referencing Students table)
 - course_id (Foreign Key referencing Courses table)
 - enrollment_date

- ...

In this setup:

- Each record in the "Enrollments" table represents a specific student enrolled in a specific course.
- The combination of "student_id" and "course_id" forms a composite primary key in the "Enrollments" table, ensuring that each student can be enrolled in a course only once.
- Both "student_id" and "course_id" columns in the "Enrollments" table are foreign keys referencing the respective primary keys in the "Students" and "Courses" tables.

One-to-One Relationship Example: Consider a scenario where each employee in a company is assigned a unique employee ID card. Each employee can have only one ID card, and each ID card is assigned to only one employee. This forms a one-to-one relationship between the "Employees" table and the "IDCards" table.

- Employees Table:
 - employee_id (Primary Key)
 - employee_name
 - ...
- IDCards Table:
 - card_id (Primary Key)
 - employee_id (Foreign Key referencing Employees table)
 - card_number
 - expiration_date
 - ...

In this setup:

- Each record in the "IDCards" table represents a unique ID card assigned to a specific employee.
- The "employee_id" column in the "IDCards" table is a foreign key referencing the primary key "employee_id" in the "Employees" table.
- Each employee can have only one corresponding record in the "IDCards" table, and each ID card is assigned to only one employee, establishing a one-to-one relationship between the two tables.


Difference between stored procedure and function -22

Function	Stored Procedure	diff
A function is a database object that accepts parameters, performs calculations or operations, and returns a single value. Functions are typically used for computations or transformations on data.	A stored procedure is a set of precompiled SQL statements that can perform a series of operations or tasks. It can contain procedural logic, such as conditional statements, loops, and error handling.	Purpose
Functions must return a single value. They can be either scalar functions (returning a single value) or table-valued functions (returning a table).	Stored procedures can return multiple result sets, modify data in the database, or perform other operations, but they do not necessarily return a value.	Return :Type
Scalar functions can be called from SQL statements, SELECT queries, WHERE clauses, and other expressions. Table-valued functions can be used in the FROM clause to return a table.	Stored procedures can be called from SQL statements or other stored procedures. They can execute DML (Data Manipulation Language) and DDL (Data Definition Language) statements.	Usage in SQL :Statements
Functions are generally not allowed to modify database state or participate in transactions. They are intended to be deterministic and have no side effects.	Stored procedures can participate in transactions. They can start, commit, or rollback transactions within their code.	Transaction :Control
Error handling within functions is limited. They cannot use TRY...CATCH blocks or handle errors in the same way as stored procedures.	Stored procedures can include error handling logic using TRY...CATCH blocks to handle exceptions and errors gracefully.	Error :Handling
Functions must return a value, either scalar or table, as specified in the function definition.	Stored procedures do not necessarily return a value. They may return output parameters or modify data in the database.	Return :Value

Difference between where and having -23

Where	Having	diff
clause is used to filter rows from the result set based on conditions applied to individual rows before any aggregate functions are calculated.	clause is used to filter groups of rows (aggregated data) based on conditions applied to the result of aggregate functions.	Usage
clause is applied to individual rows of the original data before any grouping or aggregation occurs	clause is applied to groups of rows resulting from the GROUP BY clause, after aggregation has been performed.	Applied To
clause typically appears before the GROUP BY clause (if used) and is followed by the GROUP BY clause.	clause appears after the GROUP BY clause and any aggregate functions in the SELECT statement.	Position in the Query
clause, you can use conditions to filter individual rows based on column values.	clause, you can use conditions to filter aggregated data based on the result of aggregate functions.	Filtering Conditions

sql

 Copy code

```
-- WHERE clause example
SELECT product, SUM(quantity) AS total_quantity
FROM sales
WHERE total_quantity > 10
GROUP BY product;

-- HAVING clause example
SELECT product, SUM(quantity) AS total_quantity
FROM sales
GROUP BY product
HAVING SUM(quantity) > 10;
```

CSS & html & JS

Frist CSS

-1 Difference between display block and none

Display: block عبارة عن property بقدر اديه لعنصر ال html يجعلها ك element منفرد عن باقي ال elements
يكون في سطر لوحده (new line) وياخذ الويدث كامل في مكان وجوده في الصفحة : ex :
<div>

- **Display: none** عبارة عن property بيتتم استخدامه لاختفاء وجود العنصر مع ملاحظه مسح مساحته من الصفحة والجزئية اللي تم استخدامها ليه (بمعنى كأنه لم يكن)

-2 How to make style (طرق اضافته CSS للكود وايه افضل طريقة)

يتم عمل style لصفحة ال html من خلال ال CSS ، كود ال CSS بقدر اضيفه واربطه بتاجات ال html ب 3 طرق

External: وفيها بيكون كود ال CSS في ملف خارجي بامتداد CSS. و يقوم بربطه بفايل ال html من خلال وضع تاج <link> داخل تاج ال <head>
Ex: <link rel="stylesheet" href="main.css"> < > باعتبار ان اسم الملف main.css

Internal: ودي بتكون عن طريق تاج اسمو <style> بيتتم وضعه جوا تاج ال <head> وبيتتم كتابة اكواد ال CSS مباشرة جوا التاج

Ex:

```
<head>
<style>
P {color: blue}
</style>
</head>
```

Inline: هنا كود ال CSS بيتكتب كاتريوت style جوا تاج ال html
Ex: <p style="color: red"></p> < > خلي لون البراجراف احمر

-3 تعرف ايه عن ال class في ال CSS

بشكل مختصر هي عبارة عن نوع من انواع ال selectors ، وبتتميز اني بقدر بيهاميز اكتر من element في تاج ال html فبتتيح ان يكون عندي اكتر من تاج بيحمل نفس اسم ال class وبستخدمها عشان اعطي style موحد لاكثر من عنصر في نفس الوقت
برمز لل class برمز ال dot (.)

Ex:

home <== p.home { background-color: yellow;}

-4 انواع ال position in css

- **Static**: وده ال default بتاع اي element ، ولا يتأثر العنصر بالاتجاهات بل يتواجد حسب ترتيبه في الصفحة
- **Relative**: تأثيره على العنصر انو بيخليه قادر يتحرك لاي اتجاه بحدده (left , right , ...) في حدود مكانه ي الصفحة
- **Fixed** : بتثبت العنصر في المكان اللي هحدده ليه بالنسبة لل viewport حتى لو عملت scroll للبيدج بتاعتي بيفضل ثابت
- Ex: `img{position: fixed; bottom:0px ; right:0px}` << هنا هتثبت الصورة في الاسفل ناحية الجبهه اليمين
- **Absolute** : مكان العنصر بيكون تابع للعنصر اللي قبله اللي اديت له position ، لو انا مدتش position لاي عنصر فيكون مكانه متحدد على اساس <body>
- **Sticky**: لو عندي عنصر مثلا في نص الصفحة واخد position: sticky لما بعمل scroll للبيدج وبعدي العنصر ده ، العنصر بيفضل ثابت اعلى الصفحة طول ما انا بسكرول ؛ حيث ان الخاصيه دي بتثبت العنصر مهما اتحركت لاعلى ولاسفل طول ما انا دخلت محيط ال element

-5 Difference between fixed and sticky in css

The **position: fixed** means fixed to the viewport. We provide the position values (top, bottom, right, or left) and the element stays there when the user is scrolling. No matter what is happening on screen the fixed element will not move at all.

The **position: sticky** means the element will scroll until it reaches the offset value given to it by the user and then stays in its position. Sticky element always stays within its parent block and as soon as the parent block leaves the screen as an effect of scrolling, sticky elements also leave with it.

((شرحهم بالعربي في السؤال ال 4))

-6 Difference between align item and justify content in CSS

- **justify-content**: horizontal alignment بمعنى انه هيحاذي ال item المنظور الافقي للصفحة
- **align-item**: vertical alignment بمعنى انه هيحاذي ال item المنظور العامودي للصفحة

-7 Difference between margin and padding in CSS

- **Margin** : المساحة خارج ال content اي ما بعد ال border
- **Padding** : المساحة الداخليه بين ال item وال border اي ما بداخل ال border

-8 Difference between display none and visibility hidden in CSS

- **Display-none** : تجعل ال content كأنه لم يكن ، تمحيه وتمحي المساحة التي كان يتواجد فيها تماما
- **Visibility-hidden** : تمحي ال content مع عدم حذف المساحة التي كان يأخذها من الصفحة تظل المساحة لكن يتم مسح المحتوى

-9 Difference between outline and border in CSS

- **Border** : هو الاطار الذي يحيط بال item من الداخل حيث يتم حسابه ضمن ال actual width of item
- **Outline** : هو الاطار الذي يحيط بال content كاملا من الخارج ، فلا يتم حسابه ضمن ال actual width of item

-10 What is box sizing

- CSS box-sizing enables you to define how you want the height and width of an element to be calculated. It means that you can use this property to specify whether the padding or border of the element should be included in the height and width or not.
- بمعنى اني بقدر من خلال الخاصية دي ان احدد لل content اللي جوا ال containers بتاعي ابعاده من ال Height & width خصوصا ال padding & border لكل item

-11 Difference between class and id

- كلاهم عبارة عن selectors بيستخدمو لتنظيم تنفيذ كود ال CSS على تاجات ال html
- **Class** : يستخدمه عشان اضع اسم يضم مجموعه من elements ال html بحيث لما اعمل style ليهم يتم تنفيذ ال style جمله واحده على كل ال elements التي تحمل نفس ال class ويشار اليها بعلامه ال (.)
 - **ID** : يستخدمه لوضع اسم مميز على elements واحد فقط في ال html بحيث لا يتكرر ل element ثاني في الكود كله ويشار اليه بعلامه (#)

Second HTML

1- عمل جدول بال html يحتوي على صفين وعمودين

```
<table>
<tr>
<td>first </td>
<td>first</td>
</tr>
<tr>
<td>second</td>
<td>second</td>
</tr>
```

2- Difference between div & span

- **Div**: عبارة عن block container لو عنصر اخذ الخاصيه دي بياخد المساحة كامله width ليه هوا في line منفرد
- **Span**: عبارة عن inline container لو عنصر اخذ الخاصيه دي مش بينزل line جديد بل يبقى موجود في نفس مساحة ال container اللي تم وضعه فيه

3- عمل قائمة بال html

- فيه اكثر من نوع من ال list في ال html منهم
- **Order list**: عبارة عن قائمه مرقمه ومنظمه بارقام او دلالات معينه ، وبيكون الكود بتاعها عبارة عن `` حيث `ol` هي بداية فتح القائمه ، وال `li` عبارة عن العنصر بداخل القائمه فبعدد العناصر تكون عدد تاجات `li`
- **Unorder list**: عبارة عن قائمه غير مرقممه او مرتبه ، يتم ترتيبها من خلال الرموز ال (. , * , ...) و غيرها من الرموز ، يكون الكود لتاعها كسابقتها مع تغيير بدايه التاج ل `` بدلا من ``

4- ايه هوا تاج ال mark

هو عبارة عن تاج ال output بتاعه انه بيعمل highlight على الكلمة او الجمله زي مثلا الكلمه دي

MARK

5- ايه هيا انواع ال heading في html

ال heading في ال html عبارة عن تمييز لجمله او كلمة على رأس الموضوع ويتكون واخده size معين و bold ، ويتترتب من <h1> الى <h6> الاختلاف فيما بينهم سيكون حسب حجم ال size وال bold وهكذا ف <h1> اكبر حجه نزولا الى <h6> اقل حجا

6- What is colspan & rowspan

خلال عمل جدول بلغة html تتيح لنا اللغة بدمج خليتين او اكثر سواء بشكل افقى او رأسي فتستخدم الخاصيتين ك attribute داخل كود ال html

- **Colspan** : للدمج افقيا

مثال : <tr><td colspan="3"> 3 cells </td></tr> == <= بمعنى قم بدمج 3 خلايا افقيا

- **Rowspan** : للدمج عاموديا

مثال : <tr><td rowspan="3"> 3 cells </td></tr> == <= ادمج 3 خلايا رأسيا

7- Difference between tag & attribute

- **Tags** : عبارة عن كود ينفذ لتحديد البنية الاساسيه لصفحة الويب (structure or content to webpage)

- **Attributes** : عبارة عن معلومات اضافيه او تأثيرات يتم اضافتها لل tags للتغيير من شكلها الخارجي
بمعنى اخر :

tags are like the building blocks of a webpage, while attributes provide specific instructions or details for those building blocks

Third JS

-1 What is closure in js

A closure is a function having access to the parent scope, even after the parent function has closed.

في الحالة العادية للكود لا يمكن لاي فانكشن داخلي inner برؤية متغيرات الفانكشن الخارجيه outer
لكن في حالة تطبيق closure function يمكننا بذلك الدخول ل the parent scope

للقراءة اكثر وفهمها عمليا : https://www.w3schools.com/js/js_function_closures.asp

-2 What is type of function in js

- **named functions** : الحالة العادية للفانكشن حيث تحتوي على اسم لها وعدد من parameters
- **anonymous functions**: عبارة عن فانكشن ملهاش اسم ، يستخدمها لو الكود بتاعها هيتنفذ لمره واحده او مرتين فقط خلال البرنامج ، وممكن اخزنها في متغير
- **arrow function** : عبارة عن لل syntax of anonymous function باتباع طريقه معينه
مثال : $(parameter1, parameter2) \Rightarrow parameter2 + parameter1$
** بحت البرامترز بين القوسين وبعدين بحت \Rightarrow وبعد كذا بحت ال body of function **
** بحذف كلمة return وكذلك كلمة function **

-3 In JS what is setInterval & setTimeout

اللاتين عبارة عن فانكشن جاهزة في لغة ال js بيحسبو الفترة الزمنية لل function اللي هيتم وضعها كبرامتر ولكن لكل واحد من دول output معين

- **Settimeout** : بتقوم بتنفيذ امر معين نقوم بكتابته (function) بعد وقت معين نقوم بتحديدده لمرة واحدة

- **Setinterval** : بتقوم بتنفيذ function امر معين نقوم بكتابته بعد وقت معين نقوم بتحديدده لأكتر من مره
اللاتين الوقت فيهم بيتحسب بال millisecond واللاتين من ال function التابعه لل window object

-4 What is callbackFunction in JS

عبارة عن function سيتم تنفيذها بعد تنفيذ function تانيه ، بمعنى ان ال call back function ستتخط ك argument عند تنفيذ function تانيه

a function which is to be executed after another function has finished execution. A more formal definition would be - Any function that is passed as an argument to another function so that it can be executed in that other function is called as a callback function

**** ركز ان الفانكشن ستتضاف ك argument مش كبرامتر ****

**** الفانكشن اللي ستتضاف كبرامتر بنسميها higher order function ****

-5 Difference between getElementById & getElementByName

كلاهم Dom method in JS

- **getElementById** : يتمكن من خلالها من الوصول لل element من خلال ال id بتاعه (ال id مش بيتكرر بيكون مميز لعنصر واحد بس))
- **getElementsByName** : يتمكن من خلاله من الوصول لمجموعة ال elements اللي ليها اتريبيوت اسمه name ((ال name value ممكن يتكرر لاكثر من element))

-6 How to make condition in JS

- بقدر اعمل شرط من خلال قاعده if condition او من خلال switch statement
- **If condition** : جملة شرطيه ممكن تغير من ناتج الكود على حسب الشرط المتواجد بين القوسين ، يعني لو صح نفذ كذا ولو مش صح نفذ كذا ، لأكثر من شرط يتم تطبيق else if
 - **Switch statement** : بستخدمها لو عندي اكثر من cases ممكن اليوزر يدخلها ، بمعنى اكثر من سيناريو مكن يتحقق ف بيتم الاختيار ما بين ال cases بتاعت السويتش على اساس السيناريو

-7 Difference between for loop and while in JS

- اكثر فرق جوهري بين الاثنين ان ال for loop بستخدمها لو انا عارف كام مره هقوم بتكرار الكود بتاعي
- اما ال while مش هكون عارف كام مره هيتكرر الكود بتاعي
- كمان في ال while الكود هيتنفذ عاقل مره واحده اذا كان ال condition بس true ففي حالة ال false مش هيتنفذ ابدا لكن في حالة ال for الكود بيتنفذ طول ما هوا true لحد ما يبقى الشرط false هيووقف تنفيذ
- ال for اسرع في التنفيذ من ال while

8- السؤال عموماً عن ال loop

اللوب عبارة عن مصطلح بلجاً ليه لما يكون حابب ان نفس الكود يتكرر اكثر من مره ، فبدل ما اكتب نفس الكود لأكتر من مرة ، يستخدم قاعده loop بسطر كود واحد بيتنفذ الكود لأكتر من مره فيه اكتر من قاعدة لل loop منهم

- **For** : ودي بيكون ليها بدايه و condition هيتأكد منه في كل مره تم تنفيذ الكود فيها action ويتم مع بداية التنفيذ عشان يحدد امته نهاية تكرار الكود ، وبعد كذا يكتب the block of code اللي عايز يتكرر الناتج بتاعه
- **While** : هنا بعرفلو برامتر خارج اللوب وبيكون مهمته ان ده اللي هيتحطه ال condition داخل قوسين ال while ويقدر في كود ال while اضيف شرط او break او continue
- **Do/while** : ودي باضافه ال do ل While ومهمه ال do ان بتتنفذ ايا كان الشرط true or false و لو الشرط اللي جوا قوسين ال while كان false في الحاله دي هينفذ ال do بس وهيوقف ال loop

9- What is switch in JS

عبارة عن نوع من انواع ال condition statement بقدر من خلالها انفذ حالة معينه من الكود المرتبط ب تحقيق شرط معين ممكن تنفذ مره او اكثر لانها بتعتمد عل ال cases بمعنى بتعتمد على السيناريوهات لتحقيق الشرط بتاعي لو اي سيناريو فيهم true هينفذه فبالتالي هوا هيعدي على جميع ال cases سواء وجد ال true او لا الا لو استخدمت كلمات معينه توقف بيها الكود

10- What is API

عبارة عن اهتصار ل application programming interface ، وهو عبارة عن طريقه تستخدم للتواصل بين two programs or two software component ، ف من خلاله بقدر احط ال بروتوكولز والشروط اللازمه لتواصل وتفاعل ال component مع بعضها . الويب بيستخدم ال APIs مثلاً عشان يستحضر داتا من الداتا بيز او عشان يحضر النشرة الجوية من ال weather service او مثلاً استحضار ال maps من google maps وهكذا

11- What is Angular

هي عبارة عن open-source JS front-end framework يستخدمنا عشان develop single page application بمعنى يكون كل content الويبسايت بتاعي في صفحه واحده ، ويستخدم ال html & typescript لإنشاء ده

-12 Difference between let and var in JS

ظهرت ال let كتعريف للمتغيرات في ECMAScript 6 وازافت بعض الخصائص المهم التي لم تكن موجوده في ال Var

- **كلاهم** يكون global or local حسب المكان اللي بيتحدد فيه من خلال scope لكن من ظهور ال let ظهر خاصية block scope وهو ان المتغيرات المسبوقه ب **let keyword** متبقاش متشافه برا ال block scope بمعنى مثلا لو عرفنا **let a = 1** جوا block لكود if لما اخرج برا البلوك ده مش هقدر استدعي القمه بتاعت ال a بعكس لو عرفت **var a** ساعتها بيكون عادي استدعيها وهتكون متشافه
- او عرفت نفس المتغير مرة تانيه ؛ لو كان **var** ف بقدر اني اعرف المتغير مرتين بقيم مختلفه دون التحذير من التكرار الحاصل ، لكن بال **let** بينبهنني ان فيه تكرار حاصل لأحد المتغيرات وهديلنا error
** في فرق بين اني اعرف المتغير مرتين واني اغير القيمه بتاعته **
- لو جيت مثلا وطبعت او عملت اي action للمتغير قبل ما اعرفه ؛ في حالة ال **var** هيدينا undefined ومش هيعرفنا فين الخطأ الحاصل ، لكن في حالة ال **let** بيطلعلي جمله خطأ ويوضحلي ان ده امر حاصل قبل تعريف المتغير
ودي بعض الفروقات في استخدامات ال **let & var**

-13 What is hosting

بشكل مبسط عبارة عن service بتمكنا من رفع المحتوى بتاعنا (web pages) اونلاين على الانترنت

-14 Is JS client side or server side

ال JS عبارة عن client-side بمعنى ان البراوزر هوا اللي بيعمل process لكود بتاعي قبل ما بيعته للسيرفر والطريقه دي بتستخدم عشان يتحقق من الداتا هل هي صحيحه ولا في errors وكل ده قبل ما بيعتها للسيرفر