

1- ايه هو الـ SQL؟

SQL (Structured Query Language)

دي اللغة اللي بنستخدمها عشان نكلم قواعد البيانات (Database)، سواء علشان نجيب بيانات، نضيف، نعدل أو

نحذف. 2- يعني ايه SELECT؟

SELECT دي الكلمة اللي بنبدأ بيها لما نحب نجيب بيانات من Table.

```
SELECT * FROM Customers;
```

يعني هاتلي كل الأعمدة وكل الصفوف من جدول Customers.

3- ازاي أختار أعمدة معينة؟

```
SELECT Name, Email FROM Customers;
```

ده بيحجب عمودين بس: Email و Name.

4- ازاي أفلتر النتائج؟ (WHERE Clause)

```
SELECT * FROM Customers WHERE City = 'Cairo';
```

ده بيقول: هاتلي العملاء اللي ساكنين في القاهرة بس.

5- يعني ايه DISTINCT؟

بتستخدم عشان تمنع التكرار.

```
SELECT DISTINCT City FROM Customers;
```

ده هيرجع كل المدن بس من غير تكرار

6- ازاي أرتب البيانات؟ (ORDER BY)

```
SELECT * FROM Products ORDER BY Price DESC;
```

ده بيرتب المنتجات حسب السعر من الكبير للصغير (DESC).

7- الفرق بين WHERE و HAVING؟

- WHERE بتشتغل قبل التجميع (GROUP BY)

- HAVING بتشتغل بعد التجميع

```
SELECT City, COUNT(*) FROM Customers GROUP BY City HAVING COUNT(*) > 5;
```

8- يعني ايه GROUP BY؟

بتجمع البيانات حسب قيمة معينة.

```
;SELECT City, COUNT(*) FROM Customers GROUP BY City
```

9- يعني ايه INNER JOIN؟

INNER JOIN بتجيب الصفوف المشتركة بين جدولين.

```
SELECT Orders.OrderID, Customers.Name
```

```
FROM Orders
```

```
INNER JOIN Customers ON Orders.CustomerID = Customers.ID;
```

10- الفرق بين LEFT JOIN و INNER JOIN؟

- INNER JOIN = يجيب المشترك بس
- LEFT JOIN = يجيب كل البيانات من الجدول اليسار حتى لو مفيش مقابل في الجدول الثاني

11- يعني ايه NULL؟

NULL معناها إن مفيش قيمة متخزنة في الخلية دي، ومش بصفر.

12- ازاي أتعامل مع NULL؟ (IS NULL)

```
SELECT * FROM Customers WHERE Phone IS NULL;
```

13- يعني ايه IN؟

```
SELECT * FROM Products WHERE CategoryID IN (1, 2, 3);
```

14- يعني ايه LIKE؟

```
SELECT * FROM Customers WHERE Name LIKE 'A%';
```

يجيب اللي اسمه يبدأ بحرف A.

15- الفرق بين CHAR و VARCHAR؟

- CHAR: بيحجز مكان ثابت
- VARCHAR: بيحجز مساحة حسب حجم النص الفعلي

16- يعني ايه Primary Key؟

مفتاح أساسي بيحدد الصف بشكل فريد ومينفعش يتكرر أو يبقى فاضي.

17- يعني ايه Foreign Key؟

مفتاح يربط جدول بجدول ثاني، وبيشير إلى Primary Key في الجدول الثاني.

18- يعني ايه Index؟

Index زي الفهرس بيسرّع البحث، لكن يببطّ العمليات اللي فيها Insert و Update.

19- يعني ايه Normalization؟

تنظيم الجدول علشان نقلل التكرار ونزود الكفاءة، فيه منه 1NF, 2NF, 3NF ...

20- ازاي أعمل Table؟

```
CREATE TABLE Students
(
  ID INT PRIMARY KEY,
  Name NVARCHAR(100),
  Age INT
);
```

21- ازاي أضيف صف؟

```
INSERT INTO Students (ID, Name, Age) VALUES (1, 'Ahmed', 20);
```

22- ازاي أعدّل صف؟

```
UPDATE Students SET Age = 21 WHERE ID = 1;
```

23- ازاي أ حذف صف؟

```
DELETE FROM Students WHERE ID = 1;
```

24- يعني ايه View؟

View زي جدول وهمي مبني على Query.

```
CREATE VIEW ActiveCustomers AS SELECT * FROM Customers WHERE Active = 1;
```

25- يعني ايه Stored Procedure؟

بروسيجر مخزن بتكتب فيه كود SQL وبتستخدمه كتير.

```
CREATE PROCEDURE GetCustomers AS SELECT * FROM Customers;
```

26- يعني ايه Function في SQL؟

بترجع قيمة.

```
CREATE FUNCTION GetCustomerCount() RETURNS INT AS BEGIN
```

```
RETURN (SELECT COUNT(*) FROM Customers);  
END;
```

27- يعني ايه Subquery؟

Query جوا Query

```
SELECT * FROM Products WHERE CategoryID = (SELECT ID FROM Categories WHERE Name =  
'Drinks');
```

28- يعني ايه Alias؟

بتغير اسم العمود أو الجدول في ال Output

```
SELECT Name AS CustomerName FROM Customers;
```

29- يعني ايه Transaction؟

بتستخدم في العمليات الحرجة اللي لازم تتم كلها أو تتلغى كلها.

```
BEGIN TRAN
```

```
UPDATE Accounts SET Balance = Balance - 100 WHERE ID = 1
```

```
UPDATE Accounts SET Balance = Balance + 100 WHERE ID = 2
```

```
COMMIT
```

30- يعني ايه Rollback؟

لو حصل Error، بترجع البيانات لحالتها قبل العملية.

31- يعني ايه Trigger؟

كود بيتنفذ تلقائي بعد عملية معينة زي INSERT أو DELETE.

32- يعني ايه Constraint؟

حاجة بتحط قيود على العمود زي:

NOT NULL | UNIQUE | DEFAULT

33- يعني ايه UNIQUE؟

زي ال Primary Key بس ينفع يبقى NULL.

34- يعني ايه DEFAULT؟

قيمة افتراضية للعمود لو المستخدم مدخلش حاجة.

35- ازاي أعمل Rename لعمود؟

```
EXEC sp_rename 'Students.Age', 'StudentAge', 'COLUMN';
```

36- يعني ايه UNION؟

بيدمج بيانات من أكثر من Query بشرط يكونوا نفس الأعمدة.

```
SELECT Name FROM Customers
UNION
SELECT Name FROM Employees;
```

37- الفرق بين UNION و UNION ALL؟

- UNION: يزيل التكرار
- UNION ALL: يسيب التكرار

38- ازاي أعمل Pagination؟

```
SELECT * FROM Products ORDER BY ID OFFSET 10 ROWS FETCH NEXT 10 ROWS ONLY;
```

39- ازاي أعرف عدد الصفوف؟

```
SELECT COUNT(*) FROM Orders;
```

40- ازاي أجيب أكبر قيمة؟

```
SELECT MAX(Price) FROM Products;
```

41- ازاي أجيب أقل قيمة؟

```
SELECT MIN(Price) FROM Products;
```

42- ازاي أجيب المتوسط؟

```
SELECT AVG(Price) FROM Products;
```

43- يعني ايه Schema؟

زي تصنيف للجداول داخل قاعدة البيانات.

44- يعني ايه Temp Table؟

جدول مؤقت بيتخزن مؤقتًا أثناء الجلسة.

```
CREATE TABLE #TempUsers (ID INT, Name NVARCHAR(50));
```

45- يعني ايه Clustered Index؟

هو ال Index اللي يترتب الصفوف فعليًا في القرص.

46- يعني ايه Non-Clustered Index؟

فهرس بيشار على مكان البيانات، مش يترتبها فعليًا.

47- يعني ايه ACID؟

خصائص المعاملات:

Atomicity | Consistency | Isolation | Durability

48- يعني ايه Data Type؟

هو نوع البيانات زي:

INT | VARCHAR | DATE | BIT

49- ازاي أعمل Backup؟

```
BACKUP DATABASE MyDB TO DISK = 'C:\backup.bak';
```

50- ازاي أعمل Restore؟

```
RESTORE DATABASE MyDB FROM DISK = 'C:\backup.bak';
```

51- إيه الفرق بين ال ref وال out في C#؟

الإجابة:

ال ref و out بيستخدموا علشان تبعث المتغير بال reference مش بال value، بس الفرق إن:

- ref: لازم تكون عرّفت المتغير قبل ما تبعته.
- out: مش لازم يكون متعرف، لكن لازم يتم تعيين قيمة ليه جوه الميثود.

```
void AddOne(ref int x)  
{
```

```

x += 1;
{

void CreateNumber(out int y)
}
// y = 10; لازم يتخصص قيمة
{

```

52- يعني إيه nullable types في C#؟

الإجابة: يعني النوع يقبل null، زي كده: `bool`، `int?`. مفيد لما تحب تمثل قيمة ممكن تكون موجودة أو لا.

```

int? age = null
;if (age.HasValue) Console.WriteLine(age.Value

```

53- إيه الفرق بين الـ Array و الـ List<T>؟

الإجابة:

- **Array**: ثابت الحجم، ومش بتقدر تزود عليه بعد ما تعمل initialization.
- **List<T>**: دا ديناميكي، تقدر تضيف وتمسح عناصر.

```

int[] arr = new int[3] {1,2,3};
List<int> list = new List<int>() {1,2,3};
//list.Add(4); تمام

```

54- إيه هي الـ Dictionary؟

الإجابة: نوع بيانات بيخزن قيم على شكل مفتاح وقيمة (Key-Value). مفيد للبحث السريع.

```

var myDict = new Dictionary<string, int>();
myDict["Ahmed"] = 25;
Console.WriteLine(myDict["Ahmed"]); // 25

```

55- إزاي تعمل overload لميثود في C#؟

الإجابة: يعني تعمل نفس اسم الميثود بس بأنواع أو عدد مختلف من الباراميترز.

```

void Print(string name) { }
void Print(string name, int age) { }

```

56-يعني إيه extension method؟

الإجابة: دي طريقة تزود بيها ميثود جديدة لأي type موجود من غير ما تعدل في الكود الأصلي.

```
public static class MyExtensions
{
    public static int WordCount(this string str)
    {
        return str.Split(' ').Length;
    }
}
```

57-إيه هي الـ enum؟

الإجابة: نوع بيانات يسمحك تعرف مجموعة من القيم الثابتة.

```
enum Status { Pending, Approved, Rejected };
Status s = Status.Pending;
```

58-إيه الفرق بين abstract class و interface؟

الإجابة:

- **abstract**: فيها تنفيذ جزئي وتقدر تضيف خصائص وميثودز.
- **interface**: تعريف فقط، من غير تنفيذ.

59-يعني إيه anonymous type؟

الإجابة: نوع بتعمله on-the-fly من غير ما تعرّف كلاس ليه.

```
var person = new { Name = "Ali", Age = 30 };
```

60-إيه هي var؟ وهل في مشكلة من استخدامها؟

الإجابة:

var بتخلي الـ compiler يحدد النوع تلقائيًا.

لو استخدمتها صح بتسهل الكود، بس لو استخدمتها زيادة ممكن تخلي الكود مش واضح.

61-يعني إيه LINQ؟

الإجابة:

LINQ = Language Integrated Query، أداة تكتب بيها استعلامات على الـ collections أو الـ database بشكل بسيط.


```
var names = new List<string> { "Ali", "Sara", "Mona" };  
var result = names.Where(n => n.StartsWith("A")).ToList();
```

62- الفرق بين Where و Select ؟

الإجابة:

- **Where**: فلتر العناصر.
- **Select**: تحويل العناصر لشكل جديد.

```
var even = numbers.Where(n => n % 2 == 0);  
var squares = numbers.Select(n => n * n);
```

63- يعني إيه First, FirstOrDefault, Single, SingleOrDefault ؟

الإجابة:

- **First**: يرجع أول عنصر ولو مفيش هيعمل Exception.
- **FirstOrDefault**: يرجع أول عنصر أو default لو مفيش.
- **Single**: لازم عنصر واحد بالضبط.
- **SingleOrDefault**: يرجع عنصر واحد أو default.

64- إيه الفرق بين ToList () و ToArray () ؟

الإجابة: الـ **ToList ()** يرجع List، والـ **ToArray ()** يرجع Array. النوعين بيتحولوا من IEnumerable.

65- يعني إيه deferred execution في LINQ ؟

الإجابة: يعني الاستعلام مش بيتنفذ غير لما تستهلكه فعليًا. زي لما تستخدم **foreach** أو **ToList**.

66- إزاي تعمل group في LINQ ؟

الإجابة:

```
var groups = items.GroupBy(i => i.Category);
```

67- يعني إيه Join في LINQ ؟

الإجابة: ضم جدولين ببعض بناءً على شرط معين.

```
var result = from o in orders  
join c in customers on o.CustomerId equals c.Id  
select new { o.Id, c.Name };
```

68- إيه الفرق بين Any () و All ()؟

الإجابة:

- Any(): هل في عنصر واحد على الأقل يحقق الشرط؟
- All(): هل كل العناصر بتحقق الشرط؟

69- إيه فائدة Distinct ()؟

الإجابة:

بتشيل القيم المكررة من القائمة.

70- إزاي ترتب عناصر ب LINQ؟

الإجابة:

```
var sorted = names.OrderBy(n => n);  
var sortedDesc = names.OrderByDescending(n => n);
```

71- إيه هي OOP؟ وليه بنستخدمها؟

الـ OOP هي اختصار لـ Object-Oriented Programming، يعني "البرمجة كائنية التوجه". بنستخدمها عشان نخلي الكود modular، سهل التعديل، وسهل نعيد استخدامه (Reusability).

مثال:

بنكتب كلاس اسمه Car ونقدر نعمل منه كذا عريية (Object) بمواصفات مختلفة.

72- يعني إيه Class؟

الـ Class هو Blueprint أو قالب بنوصف فيه خصائص وسلوكيات الكائن.

مثال:

```
} public class Car  
public string Color;  
} ()public void Drive  
Console.WriteLine("Driving...");  
{  
}
```

73- يعني إيه Object؟

الـ Object هو نسخة حقيقية من الـ Class.

مثال:

```
Car myCar = new Car();  
myCar.Color = "Red";
```

74- إيه الفرق بين Class و Object؟

- Class = التصميم
Object = الكيان اللي اتبنى من التصميم

75- يعني إيه Encapsulation؟

هي إننا نخبّي البيانات (Data) ونمنع التلاعب المباشر بيها، عن طريق الـ Properties والـ Access Modifiers.

مثال:

```
private int age;  
public int Age {  
    get { return age; }  
    set { if (value > 0) age = value; }  
}
```

76- يعني إيه Inheritance؟

يعني إن كلاس يرث خصائص وسلوكيات من كلاس تاني.

مثال:

```
} public class Animal  
public void Eat() => Console.WriteLine("Eating...");  
{  
  
} public class Dog : Animal  
public void Bark() => Console.WriteLine("Barking...");  
{
```

77- يعني إيه Polymorphism؟

هي إن نفس الدالة يكون ليها أشكال مختلفة.
فيه نوعين:

- Compile-time (Overloading)
- Run-time (Overriding)

78- إيه الفرق بين Overloading و Overriding؟

- Overloading: نفس اسم الميثود، بس باراميترز مختلفة.
- Overriding: نفس الميثود في كلاس الأب، وبتتغير في كلاس الابن بـ **override**.

79- مثال على Overloading:

```
public void Print(string name) {}
public void Print(int age) {}
```

80- مثال على Overriding:

```
public class Animal {
    public virtual void Speak() => Console.WriteLine("Animal sound");
}

public class Cat : Animal {
    public override void Speak() => Console.WriteLine("Meow");
}
```

81- يعني إيه Abstraction؟

هي إننا نخبي التفاصيل الداخلية، ونظهر بس المهم للمستخدم.
بنستخدم Abstract Classes أو Interfaces.

82- الفرق بين Interface و Abstract Class؟

- Abstract Class: ممكن يحتوي على كود فعلي + ممكن ترث منه كلاس واحد.
- Interface: مفيهوش كود، بس فيه Signatures، وتقدر ترث كذا واحد.

83- إيه هي Interface؟

هي عبارة عن Contract، بتحدد إيه الميثودز اللي الكلاس لازم ينفذها.

مثال:

```
interface IDriveable {  
    void Drive();  
}
```

84- إيه فائدة استخدام Interface؟

بتخلي الكود مرن، وسهل نعمل Dependency Injection، وتساعد في الـ Testability.

85- يعني إيه Constructor؟

هو ميثود خاصة بتشتغل أول ما الـ Object يتبني، وتستخدم لتهيئة البيانات.

مثال:

```
public Car(string color) {  
    this.Color = color;  
}
```

86- يعني إيه Static؟

يعني مش محتاج تبني Object عشان تستخدمها، بتخص الكلاس كله.

87- يعني إيه this؟

بتستخدم عشان تشير للكائن الحالي اللي شغال عليه الكود.

88- إزاي أعمل Property بـ Get و Set؟

```
public string Name { get; set; }
```

89- يعني إيه Access Modifiers؟

هي اللي بتحدد مين يقدر يشوف أو يعدل الميثود أو المتغير.

- public
- private
- protected
- internal

90- إيه الفرق بين private و protected؟

- **private**: مفيش كلاس ثاني يقدر يشوفه.

- **protected**: الكلاسات التي واردة بس تقدر تستخدمه.

91- إيه هي EF Core؟

EF Core دي (ORM (Object-Relational Mapper، بتربط الكود بالكود الداتابيز. يعني بتخليك تشتغل بـ C# بدل ما تكتب SQL، وهي اللي بتحوّل الأوامر تلقائيًا.

92- يعني إيه ORM؟

الـ ORM بيربط بين الـ Class والـ Table، وبين الـ Property والـ Column.

93- إيه الفرق بين EF Core و ADO.NET؟

- ADO.NET بتكتب SQL بإيدك.
- EF Core بتخليك تشتغل بكائنات (Entities) وتسبب SQL عليها.

94- إزاي أبدأ أشتغل بـ EF Core؟

1. تضيف الباكج
2. تعمل كلاس DbContext
3. تعمل الـ Entities
4. تعمل Migrations
5. تعمل Update لقاعدة البيانات

95- يعني إيه DbContext؟

ده الكلاس اللي بيربط التطبيق بالداتابيز، وفيه الـ DbSet اللي بتمثل التابلز.

مثال:

```
public class AppDbContext : DbContext
{
    public DbSet<Customer> Customers { get; set; }
}
```

96- إيه هي الـ Entity؟

هي كلاس بيمثل جدول (Table) في قاعدة البيانات.

97- إزاي أعمل Migration؟

dotnet ef migrations add InitialCreate

dotnet ef database update

98- إيه فائدة Migration؟

بتحول التغييرات اللي في الكود (model) لأوامر SQL بتحدث بيها قاعدة البيانات.

99- إزاي أعمل العلاقة One to Many؟

```
public class Department {  
  
    public int Id { get; set; }  
  
    public List<Employee> Employees { get; set; }  
  
}  
  
public class Employee {  
  
    public int Id { get; set; }  
  
    public int DepartmentId { get; set; }  
  
    public Department Department { get; set; }  
  
}
```

100- إيه الفرق بين One-to-Many و Many-to-Many؟

- One-to-Many: عنصر من جدول يرتبط بكذا عنصر من جدول تاني.
- Many-to-Many: عناصر من جدولين يرتبطوا ببعض بكذا شكل، غالبًا من خلال جدول وسيط.

101- إزاي أعمل Seed Data؟

في `OnModelCreating` بتكتب الداتا اللي هتتضاف تلقائيًا وقت عمل المايجریشن.

102- يعني إيه Lazy Loading؟

يعني الـ related data مش بتتحمل غير لما تطلبها.

بس لازم يكون الـ `Navigation Property virtual`.

103- يعني إيه Eager Loading؟

يعني تحمّل كل الداتا المرتبطة في نفس الـ Query باستخدام `Include`.

مثال:

```
_context.Orders.Include(o => o.Customer).ToList();
```

104- إيه هو ال Fluent API ؟

هو طريقة لضبط خصائص ال Entities باستخدام الكود مش Attributes.

105- إمتى أستخدم Fluent API ؟

لما عايز تحط إعدادات معقدة أو مش متاحة من خلال ال Attributes.

106- إيه الفرق بين HasKey و HasIndex ؟

- **HasKey**: لتحديد ال Primary Key
- **HasIndex**: لإنشاء Index على عمود (يساعد في الأداء)

107- إزاي أخلي عمود Not Nullable ؟

في الكود:

```
public string Name { get; set; } // Required
```

أو باستخدام:

```
builder.Property(p => p.Name).IsRequired();
```

108- إزاي أعمل Unique Constraint ؟

```
builder.HasIndex(u => u.Email).IsUnique();
```

109- إزاي أحذف Migration ؟

```
dotnet ef migrations remove
```

110- إزاي أستخدم LINQ مع EF Core ؟

عادي بتكتب كأنك شغال على List:

```
var data = _context.Users.Where(u => u.Age > 18).ToList();
```

111- يعني إيه Delegate ؟

ال Delegate هو نوع من المتغيرات بيخزن مرجع (Reference) لدالة (Method)، يعني تقدر تبعت الفانكشن كأنها متغير.

112- إمتى أستخدام الـ Delegates؟

لو عايز تبعت دالة كـ parameter، أو تنفذ أكثر من دالة في وقت معين (Multi-cast delegate).

113- إزاي أكتب Delegate؟

```
public delegate void MyDelegate(string message);
```

```
public void Show(string msg) {
```

```
    Console.WriteLine(msg);
```

```
}
```

```
MyDelegate d = Show;
```

```
d("Hello");
```

114- إيه الفرق بين Delegate و Event؟

- Delegate بيثيل دالة أو أكثر.
- Event هو شكل معين من الـ Delegate بيشتغل عند حدوث حاجة معينة (مثل الزرار اتداس عليه).

115- إيه هو الـ Action و الـ Func؟

- Action: Delegate بيرجع void.
- Func: Delegate بيرجع قيمة.

مثال Func:

```
Func<int, int, int> add = (a, b) => a + b;
```

116- إزاي أستخدام Lambda Expressions؟

هي طريقة مختصرة لكتابة الدوال (functions).

مثال:

```
Func<int, int> square = x => x * x;
```

117- إيه هو Anonymous Method؟

دالة مالهاش اسم، بتتكتب مباشرة في مكان ال Delegate.

118- إيه الفرق بين List و Array؟

- **Array**: ثابت الطول
- **List<T>**: مرنة، تقدر تضيف وتشيل عناصر بسهولة.

119- إمتي أستخدم List؟

لما تحتاج قائمة متغيرة، تقدر تعمل عليها عمليات Add, Remove, Find بسهولة.

120- إيه الفرق بين List و Dictionary؟

- **List**: بتخزن عناصر وراء بعض
- **Dictionary**: بتخزن ك Key-Value Pairs، يعني كل عنصر له مفتاح

121- إزاي أعمل Sort على List؟

```
myList.Sort();
```

ولو فيها كائنات:

```
myList.OrderBy(x => x.Name).ToList();
```

122- يعني إيه Boxing و Unboxing؟

- **Boxing**: لما تحوّل قيمة من نوع بسيط (int) إلى Object
- **Unboxing**: العكس

123- إيه الفرق بين Struct و Class؟

- **Struct** بيخزن في ال Stack
- **Class** بيخزن في ال Heap
- **Struct** أخف وأسرع بس مش مرن زي ال Class

124- إيه الفرق بين ref و out؟

- **ref**: لازم تبعت متغير متعرف قبله

- **out**: مش لازم يكون متعرف قبل، بس لازم تديه قيمة جوه الدالة

125- يعني إيه Extension Method؟

دالة بتضيف سلوك جديد لنوع موجود بدون ما تعدل عليه.

مثال:

```
public static class MyExtensions {

    public static int WordCount(this string str) => str.Split(' ').Length;

}
```

126- إيه هو Nullable Type؟

يعني تقدر تخلي نوع زي **int** يقبل **null**.

127- إيه الفرق بين == و Equals؟

- **==**: بيقارن القيم (Value) أو المرجع (Reference) حسب النوع

- **Equals**: بيقارن المحتوى حسب ما هو معرف في الكلاس

128- إزاي أستخدم foreach مع List؟

```
foreach(var item in myList) {

    Console.WriteLine(item);

}
```

129- إيه الفرق بين List و ArrayList؟

- **Generic: List<T>**, يعني Strongly Typed
- **ArrayList**: نوع قديم بيخزن أي نوع (Object)

130- يعني إيه Indexer؟

يسمحلك تتعامل مع الكلاس كأنه مصفوفة.

131- إزاي أعمل Event؟

```
public event Action OnCompleted;
```

OnCompleted?.Invoke();

132- إيه الفرق بين static و instance؟

- static: بتخص الكلاس ككل
- instance: بتخص كل كائن لوحده

133- إزاي أعرف إن List فاضية؟

```
if (!myList.Any()) { }
```

134- إيه الفرق بين First و FirstOrDefault؟

- First: هيرجع أول عنصر ولو مفيش يرمي Exception
- FirstOrDefault: هيرجع null أو default لو مفيش

135- إيه هي الـ ToList()؟

بتحوّل IEnumerable أو أي Result إلى List قابلة للتعديل.

136- يعني إيه Generic؟

بتخلي الكود قابل لإعادة الاستخدام بأنواع مختلفة.

137- إيه الفرق بين Interface و Abstract Class؟

- Interface: بس تعريف
- Abstract: تعريف + تنفيذ جزئي

138- يعني إيه Encapsulation؟

إخفاء تفاصيل التنفيذ، وتوفير واجهة استخدام بسيطة.

139- إيه الفرق بين Stack و Queue؟

- Stack: Last In First Out
- Queue: First In First Out

140- يعني إيه Dispose؟

بتستخدم عشان تفرّغ الـ Resources (زي الملفات أو الاتصال بالدايتابيز) لما تخلص منها

141- يعني إيه Middleware في ASP.NET Core؟

Middleware يعني كود بيتنفذ في النص ما بين الـ Request و الـ Response، وبتتحكم في البايبلاين

(Pipeline).

مثال:

```
app.Use(async (context, next) =>
{
    Console.WriteLine("Request");
    await next.Invoke();
    Console.WriteLine("Response");
});
```

شرح: هنا الميدل وير بتطبع قبل وبعد ما تكمل باقي البايبلاين، كأنها فلتر في السكة.

142- إيه الفرق بين AddSingleton و AddScoped و AddTransient؟

ج:

- Transient: بيتعمل Instance جديد كل مرة.
- Scoped: نفس ال Instance جوه ال Request الواحد.
- Singleton: Instance واحد طول عمر التطبيق.

مثال:

```
services.AddScoped<IMyService, MyService>();
```

شرح: كده انت ضامن إن نفس ال Service بتستخدم في كل خطوة داخل نفس ال Request.

143- يعني إيه Dependency Injection؟

ج: دي طريقة بنحقن بيها ال dependencies بدل ما نعملهم new بنفسنا.

مثال:

```
public class MyController : ControllerBase
{
    private readonly IMyService _service;

    (public MyController(IMyService service)
    {
```

```
service = service; _
```

```
{
```

```
{
```

شرح: ال Controller مش بيكرت الخدمة بنفسه، ال Framework بيديها له جاهزة.

144- يعني إيه Attribute Routing؟

ج: دي طريقة نكتب فيها ال روت فوق ال Controller أو ال Action مباشرة.
مثال:

```
[["Route("api/products]
```

```
public class ProductsController : ControllerBase
```

```
}
```

```
[["{HttpGet("{id]
```

```
;(public IActionResult Get(int id) => Ok(id
```

```
{
```

شرح: كده بنحدد إن ال GET هيروح على `api/products/5` مثلاً.

145- يعني إيه ActionResult؟

ج: ده نوع عام بترجع بيه أي نوع من الردود، زي `OK`, `NotFound`, `BadRequest`.
مثال:

```
public IActionResult GetItem(int id)
```

```
}
```

```
if (id < 1)
```

```
;(return BadRequest("ID غلط
```

```
return Ok(new { Id = id });
```

```
{
```

شرح: IActionResult نقدر نتحكم في نوع الرد حسب الحالة.

146-إزاي أستخدم Model Binding؟

ج: Model Binding بيخلي الـ ASP.NET Core يربط الـ Parameters بالـ Request تلقائيًا.
مثال:

```
[HttpPost]
```

```
(public IActionResult Create(Product model
```

```
)
```

```
){return Ok(model.Name
```

```
);}
```

شرح: الـ Product بيتبع في البادي (Body)، والـ Framework يربطه بالـ model تلقائيًا.

147-يعني إيه Model Validation؟

ج: لما تحط Data Annotations زي [Required] أو [MaxLength] عشان تتأكد من البيانات.
مثال:

```
public class Product
```

```
{
```

```
[Required]
```

```
{ ;public string Name { get; set
```

```
};
```

شرح: لو الـ Name فاضي، ModelState.IsValid هتكون False.

148-يعني إيه Filters في ASP.NET Core؟

ج: Filters بتسمحك تضيف لوجيك يتنفذ قبل أو بعد الـ Action.
مثال:

```
public class LogActionFilter : IActionFilter
```

```
{
```

```
public void OnActionExecuting(ActionExecutingContext context) => Console.WriteLine("Before  
;("Action
```

```
public void OnActionExecuted(ActionExecutedContext context) => Console.WriteLine("After  
;("Action
```

```
{
```

شرح: دي طريقة نلوج أي Action بتننفذ في ال Controller.

149- إزاي أعمل API Controller؟

ج: بستخدم [ApiController] مع Route وبتكتب Actions بيرجعوا بيانات.
مثال:

```
[ApiController]
```

```
[Route("api/[controller]")]
```

```
public class ProductsController : ControllerBase
```

```
{
```

```
[HttpGet]
```

```
;({ "public IActionResult GetAll() => Ok(new List<string> { "item1", "item2"
```

```
{
```

شرح: كده ال API جاهزة ترد على GET بـ JSON.

150- يعني إيه Content Negotiation؟

ج: ال API بتقرر ترجع JSON ولا XML حسب ال Accept header اللي جاي في ال Request.
مثال:

```
;()services.AddControllers().AddXmlSerializerFormatters
```

شرح: كده ال API تقدر ترجع XML لو الكلاينت طلب كده.

151- إزاي بستخدم FluentValidation؟

ج: دي مكتبة بتساعدك تفصل الفاليديشن عن الموديل وتكتبها بشكل منظم.
مثال:


```

<public class ProductValidator : AbstractValidator<Product
}

()public ProductValidator

}

RuleFor(p => p.Name).NotEmpty().WithMessage("Name
;(RuleFor(p => p.Price).GreaterThan(0
{
{

```

شرح: هنا بنقول إن الـ Name لازم يبقى مش فاضي، والـ Price أكبر من 0.

152- الفرق بين Controller و ControllerBase؟

ج: Controller فيه حاجات خاصة بالـ MVC زي View()، إنما ControllerBase للـ Web API فقط.
مثال:

```

{ } public class MyApi : ControllerBase

{ } public class MyMvc : Controller

```

شرح: لو API بس، استخدم ControllerBase وخلص.

153- يعني إيه DataAnnotations؟

ج: Attributes بتحطها على الـ Properties عشان تعمل فاليديشن أو تنسيق.
مثال:

```

[Required]

[(StringLength(100]

{ ;public string Name { get; set

```

شرح: هنا الـ Name لازم يكون موجود وطوله أقل من أو يساوي 100 حرف.

154-إزاي بكتب Custom Middleware؟

ج: Middleware بتعمله كلاس فيه ميثود اسمها Invoke أو InvokeAsync.
مثال:

```
public class MyMiddleware

}

;private readonly RequestDelegate _next

;public MyMiddleware(RequestDelegate next) => _next = next

(public async Task InvokeAsync(HttpContext context

}

;Console.WriteLine("دخل الميديلوير");

;await _next(context

{

{
```

شرح: بتحت الميديل وير في البايبلين بـ `app.UseMiddleware<MyMiddleware>()`.

155-إزاي أتعامل مع الـ Configuration؟

ج: ممكن تستخدم `appsettings.json` وتقرأ منه باستخدام `IConfiguration`.
مثال:

```
appsettings.json //

"MyKey": "Hello"

;["var value = Configuration["MyKey
```

شرح: كده بتقرأ القيمة اللي اسمها MyKey من الكونفيج.

156- يعني إيه Environment في ASP.NET Core؟

ج: يعني تحدد لو انت في Development ولا Production عشان تغير السلوك.
مثال:

```
(()if (env.IsDevelopment  
  
}  
  
;)app.UseDeveloperExceptionPage  
  
{
```

شرح: الصفحة دي بتظهرلك الأخطاء بشكل كامل بس في الديفيلوبمنت.

157- إزاي أتعامل مع الـ Logging؟

ج: ASP.NET Core فيه Built-in Logging بـ ILogger.
مثال:

```
public class HomeController : Controller  
  
}  
  
;private readonly ILogger<HomeController> _logger  
  
;public HomeController(ILogger<HomeController> logger) => _logger = logger  
  
()public IActionResult Index  
  
}  
  
;logger.LogInformation("دخلنا على الاندكس");  
  
;()return View  
  
{  
  
{
```

شرح: كده أي Error أو Info يتسجل في اللوجز.

158-إزاي أتعامل مع Global Exception Handling؟

ج: بتعمل Middleware مخصص للأخطاء أو تستخدم `UseExceptionHandler`.
مثال:

```
;"app.UseExceptionHandler("/Home/Error
```

شرح: أي Error يحصل هيتحول تلقائيًا على الصفحة دي.

159-يعني إيه CORS؟

ج: Cross-Origin Resource Sharing يعني تحدد مين مسموحه يطلب بيانات من الـ API.
مثال:

```
<= services.AddCors(options
```

```
}
```

```
<= options.AddPolicy("AllowAll", builder
```

```
;(())builder.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader
```

```
;/{
```

شرح: كده الـ API مفتوحة لأي دومين، بس خد بالك في البروجيكت الحقيقي لازم تقفلها.

160-يعني إيه `ActionResult<T>`؟

ج: هو نوع جديد بيرجع Response فيه داتا وكمان Status زي `ActionResult`.
مثال:

```
(public ActionResult<Product> Get(int id
```

```
}
```

```
{ "var product = new Product { Id = id, Name = "Test
```

```
;/return Ok(product
```

```
{
```

شرح: هنا بنرجع OK ومعها `Product`، لكن النوع `ActionResult<Product>` بيدي وضوح أكثر.

161-إليه الفرق بين AddSingleton و AddTransient و AddScoped؟

ج: الفرق في الـ lifetime بتاع الكائن (Object Lifecycle).
مثال:

```
services.AddScoped<IService, MyService>(); // لكل Request
```

```
services.AddTransient<IService, MyService>(); // كل مرة جديدة
```

```
services.AddSingleton<IService, MyService>(); // مره واحده بس طول عمر التطبيق
```

شرح:

- Scoped: بيتعمل instance جديد لكل request.
- Transient: بيتعمل instance جديد كل مرة تستدعيه.
- Singleton: نفس الـ instance بتفضل شغالة من أول ما الـ ابليكشن يشتغل.

162-إزاي أستخدم Dependency Injection؟

ج: بتسجل السيرفيس في Program.cs أو Startup.cs وتحقنها في الكونترولر.
مثال:

```
services.AddScoped<IMyService, MyService>();
```

```
public class HomeController : Controller
```

```
}
```

```
;private readonly IMyService _service
```

```
;public HomeController(IMyService service) => _service = service
```

```
{
```

شرح: كده السيرفيس بتاعتك بتتحقن أوتوماتيك في الكونترولر.

163-يعني إيه Tag Helper في MVC؟

ج: دي وسيلة تكتب بيها HTML بس فيها خصائص #C بتتعامل مع الداتا.
مثال:

```
<input asp-for="Name" class="form-control">
```

شرح: asp-for بيربط الـ input بالـ model الخاص بالـ Name، بيسهل الشغل بدل ما تكتبها manual.

164-إزاي أتعامل مع الـ Routing في MVC؟

ج: بتحدد Route في Program.cs أو في الـ Attributes.
مثال:

```
app.MapControllerRoute
```

```
,"name: "default
```

```
;"{"?pattern: "{controller=Home}/{action=Index}/{id
```

شرح: أي URL هيمشي على الباترن ده، يروح للكونترولر والـ action المناسبين.

165-زاي أرجع JSON من API؟

ج: بسهولة عن طريق `return Ok(object)` أو `return Json(object)` حسب الحالة.
مثال:

```
[HttpGet]
```

```
;{{ "public IActionResult GetProduct() => Ok(new { Id = 1, Name = "Milk
```

شرح: بيرجع Json بالمنتج وبيبعث 200 OK Status.

166-يه الفرق بين IActionResult و IActionResult<T>؟

ج: الاتنين بيرجعوا Response، بس `IActionResult<T>` بيوضح نوع الداتا اللي راجعة.
مثال:

```
;"public ActionResult<string> Get() => "Hello
```

شرح: هنا محدد إنك راجع string مع Status Code، وده أوضح من IActionResult بس.

167-إزاي أستخدم Filters؟

ج: Filters بتستخدم عشان تضيف logic قبل أو بعد الـ Action.
مثال:

```
public class LogActionFilter : IActionFilter
```

```
}
```

```
(public void OnActionExecuting(ActionExecutingContext context
```

```
}
```

```
Console.WriteLine("قبل الأكشن");
```

```
{
```

```
(public void OnActionExecuted(ActionExecutedContext context
```

```
}
```

```
Console.WriteLine("بعد الأكشن");
```

```
{
```

```
{
```

شرح: ممكن تستخدمها لتسجيل اللوجز أو التحقق من اليوزر أو أي logic إضافي.

168- إيه هو Model Binding؟

ج: يعني ASP.NET يربط الداتا الي جاية من الفورم أو الريكوست بال Parameters تلقائي.
مثال:

```
(public IActionResult Create(Product product
```

```
}
```

```
);return View(product
```

```
{
```

شرح: ال framework لوحده بيملا ال product من ال form fields الي جاية.

169- إزاي أعمل API Documentation؟

ج: بأداة زي Swagger الي بتركبها من NuGet وتعرضلك كل ال endpoints.
مثال:

```
dotnet add package Swashbuckle.AspNetCore
```

```
builder.Services.AddSwaggerGen
```

```
app.UseSwagger
```

```
app.UseSwaggerUI
```

شرح: بتظهرلك واجهة فيها كل ال API بشكل تفاعلي وسهل للتجربة.

170- يعني إيه FromQuery و FromBody؟

ج: **FromQuery** يعني القيم جاية من **URL**، **FromBody** يعني **JSON** أو **body** تاني.
مثال:

```
[HttpPost]
```

```
(public IActionResult Create([FromBody] Product p
```

```
)
```

```
){return Ok(p
```

```
{
```

شرح: هنا بنستقبل الـ **product** كـ **JSON** في الـ **body** مش من الـ **URL**.

171- يعني إيه Model Validation؟

ج: يعني تتأكد إن الداتا اللي داخلة لـ **API** مضبوطة قبل ما تكمل المعالجة.
مثال:

```
} public class ProductDto
```

```
{[Required]
```

```
{ ;public string Name { get; set
```

```
{
```

```
[HttpPost]
```

```
){public IActionResult Create(ProductDto dto
```

```
{if (!ModelState.IsValid
```

```
){return BadRequest(ModelState
```

```
);return Ok(dto
```

```
{
```

شرح: لو الـ **Name** فاضي، **ModelState** هترجع **false** وهترجع **BadRequest**.

172- إيه فائدة FluentValidation؟

ج: بتخليك تكتب الـ Rules بطريقة مفصلة عن الموديل وبـ DSL سهلة.
مثال:

```
<public class ProductValidator : AbstractValidator<ProductDto
{
    ()public ProductValidator
    {
        ;("RuleFor(x => x.Name).NotEmpty().WithMessage("Name required
        {
        {
```

شرح: بدل ما تحط [Required] في الموديل، بتحطها في validator class وبتربطها في DI.

173- يعني إيه Middleware؟

ج: دي قطعة كود بتمر على كل request response، ممكن تضيف logic زي logging, auth، إلخ.
مثال:

```
<= (app.Use(async (context, next
{
    ;("قبل")Console.WriteLine
    ;()await next
    ;("بعد")Console.WriteLine
    ;{
```

شرح: الكود ده بيكتب log قبل وبعد أي request يدخل السيرفر.

174- إزاي أستخدام Attribute Routing؟

ج: بتكتب [Route("...")] فوق الأكشن أو الكونترولر.
مثال:

```
["Route("api/products]

} public class ProductController : ControllerBase

["{HttpGet("{id]

;(public IActionResult Get(int id) => Ok(id

{
```

شرح: لو كتبت api/products/5/ هيدخلك على الأكشن ده ويدي id = 5.

175- يعني إيه DTO وليه بستخدمه؟

ج: ده اختصار لـ Data Transfer Object، بنفصل بيه بين الكيانات واللي بترجعه للمستخدم.
مثال:

```
} public class Product

{ ;public int Id { get; set

{ ;public string Name { get; set

{ ;public decimal Cost { get; set

{

} public class ProductResponseDto

{ ;public string Name { get; set

{
```

شرح: بنخلي اللي يوصل للمستخدم أقل ومحدد، علشان الأمان والبساطة.

176- إزاي أعمل return لحالة NotFound؟

ج: باستخدام return NotFound() أو return NotFound("message").
مثال:

```
;("if(product == null) return NotFound("Product not found
```

شرح: لو الحاجة مش موجودة في الداتا، رجّع كود 404 للمستخدم.

177-إزاي أستقبل براميتر من ال Route؟

ج: عن طريق `{}` في المسار، وبعدين تستقبله ك براميتر في الميثود.
مثال:

```
[("{HttpGet("/{id}
```

```
;(public IActionResult Get(int id) => Ok(id
```

شرح: الراوت اللي فيه `api/product/2/` يروح للأكشن ده ويدي `id = 2`.

178-إزاي أعمل Error Handling عام؟

ج: بوسيلة زي ال `UseExceptionHandler` أو `middleware` مخصص.
مثال:

```
;"app.UseExceptionHandler("/error
```

```
[("Route("error]
```

```
;(public IActionResult Error() => Problem
```

شرح: أي error بيحصل بيتم توجيهه للمسار ده وتتعامل معاه زي ما تحب.

179-إيه الفرق بين 204 و 200 في ال Status Code؟

ج: 200 معناها نجاح وفيه داتا، 204 معناها نجاح لكن مفيش داتا.
مثال:

```
return NoContent(); // => 204
```

شرح: بنستخدمها مثلاً بعد update ناجح بس مش محتاج ترجع حاجة.

180-إزاي أستقبل ملفات Upload في API؟

ج: باستخدام `IFormFile`.

مثال:

```
[HttpPost]
```

```
} (public IActionResult Upload(IFormFile file
```

```
;(var path = Path.Combine("Uploads", file.FileName
```

```
;(using var stream = new FileStream(path, FileMode.Create
```

```
;file.CopyTo(stream
```

```
;())return Ok
```

```
{
```

شرح: يرفع الملف ويحفظه في فولدر على السيرفر.

181-إزاي أستخدم Filters على مستوى المشروع كله؟

ج: بتسجلها في `services.AddControllers(options` <= ...).

مثال:

```
<= services.AddControllers(options
```

```
}
```

```
;())options.Filters.Add(new LogActionFilter
```

```
;{
```

شرح: كده الفلتر ده هيتطبق على كل الكنترولرز من غير ما تكتبه عليهم.

182-إزاي أعمل Pagination؟

ج: باستخدام Skip و Take من LINQ.

مثال:

```
;()var items = db.Products.Skip((page - 1) * size).Take(size).ToList
```

شرح: لو عندك صفحة رقم 2 وحجم الصفحة 10، هيتخطى أول 10 ويأخذ التانيين.

183-إيه هو `ActionResult<T>` وليه أستخدمه؟

ج: بيرجع داتا ونوع ال response في نفس الوقت، وبيخلي الكود واضح أكثر.

مثال:

```
;()public ActionResult<Product> Get(int id) => _context.Products.Find(id
```

شرح: ممكن ترجع المنتج أو ترجع NotFound، كلها في نفس الريميتيرن.

184-يعني إيه Status Codes؟

ج: أكواد بتوصف حالة ال Response زي 200 OK, 401 Unauthorized, 404 Not Found.

مثال:

```
;"return StatusCode(500, "Server error
```

شرح: كده بنرجع رسالة خطأ 500 للمستخدم.

185-إزاي أخلي Controller خاص بـ API؟

ج: تستخدم ControllerBase بدل Controller، ومتحطش View.

مثال:

```
[ApiController]
```

```
[Route("api/[controller]")]
```

```
public class ProductsController : ControllerBase
```

شرح: ده بيخلي الكنترولر شغال كـ API فقط من غير Views.

186-إزاي أعمل Versioning لـ API؟

ج: باستخدام باكيج Microsoft.AspNetCore.Mvc.Versioning.

مثال:

```
dotnet add package Microsoft.AspNetCore.Mvc.Versioning
```

```
<= services.AddApiVersioning(config
```

```
}
```

```
;(config.DefaultApiVersion = new ApiVersion(1, 0
```

```
;config.AssumeDefaultVersionWhenUnspecified = true
```

```
);{
```

شرح: كده ممكن تعمل API V1 و V2 وتخلي الكلاينت يختار أنهي واحدة يشتغل عليها.

187-إزاي أستخدم Authorization؟

ج: تحط [Authorize] فوق الكنترولر أو الأكشن.

مثال:

```
[Authorize]
```

```
[HttpGet]
```

```
;"public IActionResult GetSecureData() => Ok("Secret
```

شرح: لازم المستخدم يكون عامل Login ومعه Token علشان يدخل.

188-إزاي أسمح لناس معينين بس يدخلوا؟

ج: باستخدام Roles مع ["Authorize(Roles = "Admin]

مثال:

```
["Authorize(Roles = "Admin]
```

```
;"public IActionResult OnlyAdminCanSee() => Ok("Welcome Admin
```

شرح: لو المستخدم مش Admin، مش هيقدر يدخل هنا.

189-إيه هو JWT؟

ج: JSON Web Token، طريقة آمنة تبعت بيها معلومات المستخدم في التوكن بعد تسجيل الدخول.

مثال:

```
}
```

```
,"alg": "HS256"
```

```
"typ": "JWT"
```

```
{
```

شرح: التوكن بيبقى فيه بيانات زي ID و Role، وبيتبعت في Header لكل request.

190-إزاي أضيف CORS؟

ج: تفعّل CORS لو عايز تسمح لـ Frontend يكلم الـ API من دومين تاني.

مثال:

```
<= builder.Services.AddCors(options
```

```
}
```

```
;(options.AddPolicy("AllowAll", p => p.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader
```

```
;{{
```

```
;"app.UseCors("AllowAll
```

شرح: كده الـ API بييسمح لأي ويب أب يشغّل عليه request.

191-عني إيه HTML؟

ج: HTML هي اللغة اللي بنبني بيها الهيكل الأساسي لأي صفحة ويب (Structure).
مثال:

```
<h1>Welcome to my site</h1>
```

شرح: العنصر <h1> ده بيعرض عنوان كبير في الصفحة.

192-يعني إيه CSS؟

ج: CSS هي اللي بنظبط بيها الشكل والستايل (Style) بتاع عناصر الـ HTML.
مثال:

```
<p style="color: red;">Hello</p>
```

شرح: السطر ده هيعرض كلمة "Hello" باللون الأحمر.

193-يعني إيه JavaScript؟

ج: JavaScript هي اللغة اللي بتحرك وبتخلي الصفحة تفاعلية (Dynamic Behavior).
مثال:

```
<button onclick="alert('Hi')">Click Me</button>
```

شرح: لما تضغط الزرار، بيظهرلك رسالة فيها "Hi".

194-يعني إيه Bootstrap؟

ج: Bootstrap هو CSS framework جاهز بستايلات وكلاسات بتسهّل عليك التصميم.
مثال:

```
<button class="btn btn-primary">Save</button>
```

شرح: الزرار ده بياخد شكل أزرق جذاب بلمسة واحدة.

195-يعني إيه TypeScript؟

ج: TypeScript هي JavaScript لكن أقوى لأنها بتدعم أنواع البيانات (Type safety).
مثال:

```
;let age: number = 25
```

شرح: هنا بنقول إن المتغير لازم يكون رقم، وده بيقلل الأخطاء وقت الكتابة.

196- يعني إيه Element في HTML؟

ج: Element هو أي حاجة مكتوبة بين < > في HTML.
مثال:

```
<p>Hello World</p>
```

شرح: العنصر <p> ده اسمه Paragraph Element.

197- إزاي أغير لون النص باستخدام CSS؟

ج: ممكن تستخدم color في ال style.
مثال:

```
<p style="color: green;">Success</p>
```

شرح: بيعرض الجملة "Success" بلون أخضر.

198- إزاي أستخدم class في CSS؟

ج: بتحط class في HTML وتكتب الستايل في CSS.
مثال:

```
<p class="danger">Warning!</p>
```

```
<style>
```

```
{ ;danger { color: red.
```

```
<style/>
```

شرح: class اسمها danger بتخلي النص لونه أحمر.

199- إزاي أضيف صورة؟

ج: باستخدام العنصر وتحدد ال src.
مثال:

```
<"img src="logo.png" alt="Logo">
```

شرح: الصورة بتظهر من الملف logo.png ولو مش موجودة بيظهر alt text.

200- إزاي أعمل لينك؟

ج: باستخدام .
مثال:

Go to Google

شرح: بيحولك لموقع جوجل لما تضغط.

201-إزاي أكتب كومنت في HTML؟

ج: بتحت الكومنت بين <!-- -->.

مثال:

<!-- دي فقرة عن الموقع -->

<p>Welcome</p>

شرح: السطر الأول مش بيظهر في الصفحة، بس بيساعد المبرمج.

202-يعني إيه div؟

ج: div عنصر حاوي بيجمع عناصر تانية جوام.

مثال:

<div>

<h2>Title</h2>

<p>Paragraph</p>

</div>

شرح: بنستخدمه ك Container لـ layout.

203- إزاي أعمل زرار؟

ج: باستخدام <button> أو <input type="button">.

مثال:

<button>Click</button>

شرح: بيعرض زرار عادي عليه كلمة "Click".

204-يعني إيه inline style؟

ج: Style مكتوب جوه العنصر نفسه.

مثال:

<h1 style="font-size: 30px;">Big Title</h1>

شرح: يخلي الخط كبير لأنه حاطط style داخلي.

205- يعني إيه ID في HTML؟

ج: id هو اسم فريد للعنصر، ممكن تستخدمه في JS أو CSS.

مثال:

```
<p id="msg">Hello</p>
```

```
<script>
```

```
;"document.getElementById("msg").innerText = "Hi"
```

```
</script>
```

شرح: غيرنا النص داخل العنصر باستخدام ID.

206- إزاي أستخدم external CSS؟

ج: بربط ملف CSS خارجي باستخدام <link>.

مثال:

```
<"link rel="stylesheet" href="style.css">
```

شرح: كده الصفحة بتاخذ الستايل من style.css.

207- إزاي أخلي النص في النص؟

ج: استخدم text-align: center;

مثال:

```
<p style="text-align: center;">Centered</p>
```

شرح: الكلام بيتوسط في العنصر.

208- إيه الفرق بين var و let في JS؟

ج: var قديم وواسع النطاق، let حديث ومحكوم بالنطاق (block scoped).

مثال:

```
;let x = 10
```

شرح: x متاحة بس داخل البلوك الي اتعرّفت فيه.

209-إزاي أعمل function في JS؟

ج: تكتب function واسمها وجسمها.

مثال:

```
} ()function sayHi  
  
;("alert("Hi  
  
{
```

شرح: بتعرض رسالة "Hi" لما تستدعي الفانكشن.

210-إزاي أستخدام Bootstrap في الصفحة؟

ج: تضيف لينك الـ CDN في <head>.

مثال:

```
link href="https://cdn.jsdelivr.net/npm/bootstrap/dist/css/bootstrap.min.css">  
<"rel="stylesheet
```

شرح: الصفحة تأخذ الستايلات الجاهزة من Bootstrap مباشرة.

211-يعني إيه container في Bootstrap؟

ج: عنصر يبطب الصفحة ويخلي فيها margins تلقائيًا.

مثال:

```
<"div class="container">  
  
<h1>Hello</h1>  
  
<div/>
```

شرح: العنصر جوهر container بيتحرك تلقائيًا في النص.

212-إزاي أعمل button أحمر في Bootstrap؟

ج: تستخدم الكلاس btn btn-danger.

مثال:

```
<button class="btn btn-danger">Delete</button>
```

شرح: الزرار لونه أحمر ويمثل عملية خطيرة زي حذف.

213-يعني إيه Responsive Design؟

ج: التصميم بيتغير حسب حجم الشاشة (موبايل /كمبيوتر).
مثال:

```
<"meta name="viewport" content="width=device-width, initial-scale=1.0">
```

شرح: السطر ده بيخلي الموقع يتظبط على الموبايل كويس.

214-إزاي أكتب شرط بسيط في JavaScript؟

ج: باستخدام if.

مثال:

```
} (if (age > 18  
;("alert("Adult  
{
```

شرح: لو السن أكبر من 18، ه يظهر رسالة إن الشخص بالغ.

215-إزاي أعرف نوع متغير في TypeScript؟

ج: تكتب النوع بعد النقطتين.

مثال:

```
;"let name: string = "Ali
```

شرح: كده TypeScript مش هيقبل غير string في المتغير ده.