

AI-ML Fresher Interview Notes (Full Detailed Version)

L1 — Technical Fundamentals

Round checks your basics in programming and problem-solving.

1. Programming Fundamentals

Variables

A name given to a memory location where data is stored.

Value can change during execution.

Example:

```
age = 22  
name = "John"  
salary = 45000.50
```

Data Types

Type	Example	Use
int	x = 10	Count, indexing
float	pi = 3.14	Decimal values
string	"AI ML"	Text data
bool	True / False	Conditions
list	[1,2,3]	Store collections
tuple	(1,2,3)	Fixed data
dict	{"name": "Amit"}	Key-value pairs

Control Statements (Decision-Making)

Example:

```
if score > 35:  
    print("Pass")else:  
    print("Fail")
```

Loops (Repetition)

Example:

```
for i in range(5):
    print(i)
while x < 10:
    x += 1
```

Operators

Arithmetic: + - * / %

Logical: and or not

Relational: > >= == !=

Strings (Immutable)

Common operations:

```
s = "Machine Learning"
print(len(s))      # length
print(s.lower())   # lowercase
print(s.upper())   # uppercas
eprint(s[0:7])     # slicing
print(s.replace("Learning", "Vision"))
print(s.split(" ")) # ['Machine','Learning']
```

Functions (Reusable Code)

Example:

```
def add(a, b):
    return a + b
print(add(3,5))
```

Parameters → inputs to a function

Return → sends output back

2. Object-Oriented Programming (OOP)

Concept	Purpose	Example
Class	Template/blueprint	Car, Student
Object	Real-world instance	myCar = Car()
Encapsulation	Protect data	private attributes
Inheritance	Reuse parent features	class Dog(Animal)
Polymorphism	Same method, different behavior	speak() for dog/cat
Abstraction	Hide complexity	ATM UI hides code

Example:

```
class Animal:  
    def sound(self):  
        print("Animal sound")  
class Dog(Animal): # Inheritance  
    def sound(self): # Polymorphism  
        print("Bark!")
```

```
obj = Dog()  
obj.sound()
```

3. Data Structures & Algorithms

Data Structures

Structure	Working	Use Case
Array/List	Sequential storage	Searching, iteration
Stack	LIFO: push/pop	Undo, Call stack
Queue	FIFO: enqueue/dequeue	Scheduling
Linked List	Node next pointer	Dynamic storage

Example Stack:

```
stack = []  
stack.append(10)  
stack.append(20)  
stack.pop() # 20
```

Searching Algorithms

Method	Steps	Time
Linear	Check all elements	$O(n)$
Binary	Repeatedly divide sorted array	$O(\log n)$

Binary search example:

`mid = (start + end)//2`

Sorting Algorithms

Sort	Steps	Time
Bubble	Swap adjacent	$O(n^2)$
Selection	Pick min every time	$O(n^2)$
Insertion	Insert into sorted area	$O(n^2)$

Example Bubble Sort concept:

Loop through array, swap if out of order

4. Time & Space Complexity (Big-O)

Big-O	Meaning	Example
$O(1)$	Constant	Direct index access
$O(\log n)$	Fast	Binary search
$O(n)$	Linear	Linear search
$O(n^2)$	Slow	Bubble sort

Used to compare **efficiency of algorithms**

5. Common Problem Questions with Examples

✓ Reverse String:

`s[::-1]`

✓ Palindrome Check:

`s == s[::-1]`

✓ Count Frequency:

```
from collections import Counter  
Counter(arr)
```

✓ Remove Duplicates:

```
list(set(arr))
```

✓ Find max:

```
max(arr)
```

Practice → **HackerRank, LeetCode Easy**

L2 — Advanced + Practical Interview Notes

This round checks whether you can **apply your technical knowledge** to real-world scenarios.

1. Full-Stack Basics (Awareness Only)

Even for an AI-ML role, companies want you to understand **how ML models are deployed** → web apps, APIs, databases.

Frontend Basics

Tech	What it is	Why needed	In AI-ML
HTML	Structure of webpage	Defines UI elements	Display UI (text, result of predictions)
CSS	Styling	Colors, spacing, layouts	Make dashboards & results attractive
JavaScript	Logic on browser	Handle user events, call APIs	Interact with ML APIs, charts

Example: Simple HTML Button

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

Example: Calling backend API using fetch()

```
fetch("/predict", {  
  method: "POST",  
  body: JSON.stringify({data: input}),  
  headers: {"Content-Type": "application/json"}  
})  
.then(res => res.json())  
.then(data => console.log(data));
```

Interview Qs (Examples)

- ✓ What is DOM?
- ✓ Difference: <div> vs
- ✓ What is fetch API?

Backend Basics

You only need basic idea — backend handles **logic + ML prediction**.

Popular frameworks:

Language Framework

Python **Flask, Django**

JavaScript Node.js

Java Spring Boot

Backend tasks:

Receive request from frontend

Run ML model and send result back

Example: Flask API for ML Prediction

```
from flask import Flask, request, jsonifyimport joblib
```

```
app = Flask(__name__)  
model = joblib.load("model.pkl")  
@app.route("/predict", methods=["POST"])def predict():  
    data = request.get_json()  
    result = model.predict([data["features"]])  
    return jsonify({"prediction": int(result[0])})  
  
app.run()
```

REST API Concepts

Term	Meaning	Example
Endpoint	URL to access a resource /predict	
HTTP Methods	Type of operation	GET, POST, PUT, DELETE
Status Code	Result of call	200 OK, 404 Not Found

Example Request Flow (Frontend → Backend)

User enters text → JS → POST → Flask → ML Model → Response JSON

Common Interview Qs

What is REST API?

GET vs POST difference?

What is JSON?

2. Databases (SQL)

Why Databases in AI?

Store datasets

Store user feedback/predictions

ML pipeline logs

Must-Know Concepts

Concept	Description
Table	Rows (records) + Columns (fields)
Primary Key	Unique identifier (ID)
Foreign Key	Links tables

Basic SQL Queries

Select with condition

```
SELECT name, age FROM students WHERE age > 20;
```

Join example

```
SELECT o.order_id, c.name FROM orders o JOIN customers c ON  
o.customer_id = c.id;
```

Grouping Example

```
SELECT department, COUNT(*) FROM employees GROUP BY department;
```

Normalization

Goal: avoid duplicate data

Levels:

1NF: atomic values

2NF: full dependency on key

3NF: no transitive dependency

Interview Qs

Why normalization?

Primary key vs foreign key?

Write a query to fetch top 5 salaries.

3. Cloud Fundamentals

Why Cloud in AI?

Training large models needs GPUs

Hosting APIs and ML Web apps

Common Services

Category	AWS Example	Use
Compute	EC2	Run ML apps
Storage	S3	Store datasets
Database	RDS	SQL database

Category	AWS Example	Use
ML Service	SageMaker	Train/Deploy models

Cloud Service Models

Model	You Manage	Provider Manages	Example
IaaS	Apps + OS	Hardware	AWS EC2
PaaS	Only apps	OS + Infrastructure	Heroku
SaaS	Nothing	Everything	Gmail

Interview Qs

What is cloud computing?

Advantages: scalability, cost control

4. Git & DevOps Basics

Version Control Concepts

Term	Meaning
Commit	Save version
Branch	Parallel development
Merge	Combine changes
Push/Pull	Sync with GitHub

Basic Git Commands

```
git init      # create repo
git add .     # stage files
git commit -m "msg"  # save snapshot
git push      # upload to GitHub
git pull      # fetch updates
```

Interview Qs

Why use Git in teams?

What is conflict and how to resolve?

CI/CD (Concept Only)

CI: Test + build automatically

CD: Deploy automatically

Helps ML pipelines update continuously.

5. AI-ML Awareness (Important Area)

Differences

Term	Scope
AI	Broad — mimic intelligence
ML	Learn from data (subset of AI)
DL	Neural networks (subset of ML)

ML Workflow

- 1 Collect dataset
- 2 Split → Train / Test
- 3 Choose model
- 4 Train (learn patterns)
- 5 Evaluate (accuracy, RMSE...)
- 6 Deploy (API/Cloud)

Key Concepts

Term	Meaning	Example
Feature	Input variables (X)	Height, weight
Label	Output variable (y)	BMI
Classification	Predict category	Spam/Not spam
Regression	Predict number	House price
Clustering	Group without labels	Customer segmentation

Common Algorithms & Use Cases

Type	Algorithm	Use Case
Regression	Linear Regression	Predict salaries
Classification	Logistic Regression, SVM, Random Forest	Fraud detection
Clustering	K-Means	Group customers
DL	CNN, RNN	Image/Text data

Example: Classification using Logistic Regression (Concept)

```
from sklearn.linear_model import LogisticRegression  
model = LogisticRegression()  
model.fit(X_train, y_train)  
prediction = model.predict(X_test)
```