HOW N8N HANDLES DATA

🎯 What you'll learn:

Understand how data flows through n8n using JSON, how nodes process it, and how to use expressions and transformations to build clean, reliable workflows.

📦 JSON & Lists in n8n

- **n8n uses JSON** to structure all data in **key-value pairs**
- Example: **{ "name": "Alice", "email": "alice@email.com" }**
- A **list** is a collection of JSON objects → like rows in a spreadsheet
- Example: 3 contacts = 3 items in a list

⚙️ How Nodes Process Data

- Each **JSON object = one item**
- Nodes **process one item at a time**
- Example: 3 items → node runs 3 times
- Always keep track of how many items are being processed

📝 Fixed vs. Expression Fields

- Fixed field: Static text (e.g. "Welcome!")
- Expression field: Dynamic value from data **({{ $json.name }})**

💡 **Look for the [*ƒ*] icon** in fields to switch between fixed and dynamic.

Best practice:

Use expressions when values change per item → keep workflows flexible.

💬 Expressions: Access & Modify Data

- Used to **read or change data** dynamically
- Example to access email: **{{ $json.email }}**
- Combine fields: **{{ $json.firstName + ' ' + $json.lastName }}**

Best practice:

Avoid hardcoding values. Always use expressions when possible.

HOW N8N HANDLES DATA

🔄 Data Transformation Tools

- **Set Node** – Rename, remove, or add simple fields
- **Function Node** – Write small JavaScript snippets
- **Code Node** – For complex custom logic

Example: Convert to uppercase

- With expression: **{{ $json.name.toUpperCase() }}**
- With code:

return [{ json: { name: $json.name.toUpperCase() } }];

Best practice:

Start simple. Use code only when expressions/set node can't handle the logic.

📂 Nested JSON Data

- APIs often return nested JSON like:

{"user": {"contact": {"email": "me@email.com"}}}

- To access nested values: {{ $json.user.contact.email }}
- **Fix structure early** using a Set node if needed

🔁 Real Workflow Structure

- **Trigger** – Start (webhook, manual, etc.)
- **Data retrieval** – From Sheets, APIs, etc.
- **Processing** – Format, filter, transform
- **Output** – Send data to Slack, email, etc.

Example:

3 rows from Google Sheets → filtered → 2 rows → sent to Slack

📥 Input & Output Tips

- If a node fails, it's often due to **wrong input format**
- Example: If email is inside **data**, use **{{ $json.data.email }}**
- **Always check execution data** of previous nodes

n8n
MASTERCLASS