



## N8N: The AI Workflow Automation



@ai.setayesh



@\_76shayan

# The Content

- 01 Why N8N?
- 02 First Installation and work with N8N
- 03 Programming Basics
- 04 Variables, Conditions and Decision-Making
- 05 What is JSON?
- 06 Work with N8N nodes in action!

**n8n یک ابزار ساده برای خودکار سازی کار های تکراریه؛**

**مثلا میتونی باهاش ایمیل های هر روز تو چک کنی، یا خلاصه  
کنی ایمیل هاتو یا ربات تلگرام بنویسی تا کاری که میخوای رو به  
صورت خودکار واست انجام بده!**



یکی از بزرگ‌ترین مزیت‌های n8n این است که با رابط کاربری گرافیکی و قابل فهم، به شما این امکان را می‌دهد که فقط با کشیدن و رها کردن (drag & drop) اجزای مختلف، فرآیندهای پیچیده را بسازید.

n8n با پشتیبانی از صدها سرویس مختلف، دستیار دیجیتالی شما می‌شود تا وقتتان را از کارهای خسته‌کننده آزاد کنید و روی بخش‌های مهم‌تر تمرکز کنید.

## نقاط قوت N8N

**متن باز (Open Source)** – می‌تونی کد منبع رو ببینی، تغییر بدی یا شخصی‌سازی کنی  
**Self-hosted بودن** – می‌تونی روی سرور خودت نصبش کنی و کنترل کامل روی داده‌ها داشته باشی.  
**بدون محدودیت در تعداد Workflow و اجراها** – برخلاف Zapier که در پلن رایگان محدودیت داره.  
**قابلیت توسعه پذیری بالا** – می‌تونی Nodeهای سفارشی خودت رو بنویسی و اضافه کنی.

## نقاط ضعف N8N

**نیاز به راه‌اندازی اولیه** – برخلاف Zapier، برای استفاده باید خودت n8n رو نصب و پیکربندی کنی (مخصوصاً اگر Self-hosted باشه).  
**نیاز به دانش پایه‌ای از مفاهیم تکنیکی مثل JSON و API** – برای بعضی نودها (مثلاً HTTP Request) بهتره بدونی API چی هست.  
**رابط کاربری برای کاربران کاملاً غیر فنی ممکنه اولش کمی گیج‌کننده باشه** – چون قابلیت‌های زیاد و تنظیمات پیشرفته داره.

## نقاط قوت Zapier

**راه‌اندازی سریع و بدون نیاز به نصب** – فقط ثبت‌نام می‌کنی و بلافاصله شروع به ساخت اتوماسیون می‌کنی.  
**رابط کاربری ساده و کاملاً کاربرپسند** – مناسب برای افراد با دانش بسیار پایین فنی.  
**تست و اجرای زنده بسیار سریع** – راحت می‌تونی ببینی هر مرحله دقیقاً چه خروجی داده.  
**اتصال به بیش از ۵۰۰۰ سرویس مختلف** – پوشش‌دهی گسترده سرویس‌ها و ابزارهای SaaS شناخته‌شده.

## نقاط ضعف Zapier

**محدودیت‌های جدی در پلن رایگان** – مثلاً تعداد Workflow کم، نداشتن قابلیت شرط، تأخیر زمانی در اجرای Task و...  
**روند قیمت‌گذاری نسبتاً گرونه** – برای استفاده واقعی و حرفه‌ای باید هزینه ماهانه/سالانه بدی.  
**امکان شخصی‌سازی و توسعه‌ای محدود** – نمی‌تونی Node سفارشی اضافه کنی یا سیستم رو تغییر بدی.



## 1. n8n.cloud (سرویس ابری رسمی)


- **توضیح:** n8n خودش یک نسخه ابری ارائه داده که نیاز به هیچ نصب و راه اندازی ای ندارد. فقط ثبت نام می کنی و وارد محیط می شی.
- **مزایا:** سریع ترین راه برای شروع؛ نیاز به نصب نیست؛ همیشه در دسترس.
- **معایب:** رایگان نیست (بعد از دوره آزمایشی، باید پلن بخری)؛ کنترل کامل روی داده ها نداری.
- **مناسب برای:** کسانی که می خوان فقط n8n رو تست کنن یا نمی خوان درگیر سرور و نصب بشن.

## 2. نصب با Docker (پیشنهادی برای Self-hosted)

- **توضیح:** رایج ترین و پایدارترین روش برای راه اندازی n8n روی سرور شخصی. فقط با چند خط دستور Docker، بالا میاد.
- **مزایا:** قابل اجرا روی هر سرور لینوکسی یا حتی ویندوزی؛ می تونی نسخه دقیق رو مشخص کنی؛ قابل نگهداری و بکاپ گیری راحت.
- **معایب:** نیاز به دانش اولیه Docker؛ باید برای نگهداری و آپدیتش برنامه ریزی داشته باشی.
- **مناسب برای:** کسانی که می خوان n8n رو دائمی روی سرور خودشون داشته باشن.



## مرحله ۱ – نصب Docker

برای ویندوز و مک‌اواس: 

1. برو به سایت <https://www.docker.com/products/docker-desktop>
2. دکمه آبی‌رنگ **Download for Windows/Mac** رو بزن.
3. فایل نصب رو اجرا کن و طبق راهنمای نصب پیش برو (Next → Next)
4. بعد از نصب، سیستم رو ری‌استارت کن و برنامه **Docker Desktop** رو باز کن.
5. صبر کن تا علامت نهنگ آبی در Taskbar ظاهر شه و سبز بشه

## مرحله ۲ – تست نصب Docker

ترمینال (یا CMD) رو باز کن و بزن:

```
~ docker --version
Docker version 28.3.3, build 980b856
~
```



## مرحله ۳ – اجرای n8n با یک دستور

حالا وقتشه n8n رو نصب کنیم:

توضیح ساده:

- `p 5678:5678`: پورت n8n رو روی سیستم باز می‌کنه
- `v ~/.n8n:/home/node/.n8n`: داده‌ها رو روی سیستم ذخیره می‌کنه تا با هر بار اجرا از بین نرن
- `n8nio/n8n`: آدرس نسخه اصلی n8n از Docker Hub

```
~ docker run -it --rm \
-p 5678:5678 \
-v ~/.n8n:/home/node/.n8n \
n8nio/n8n
```



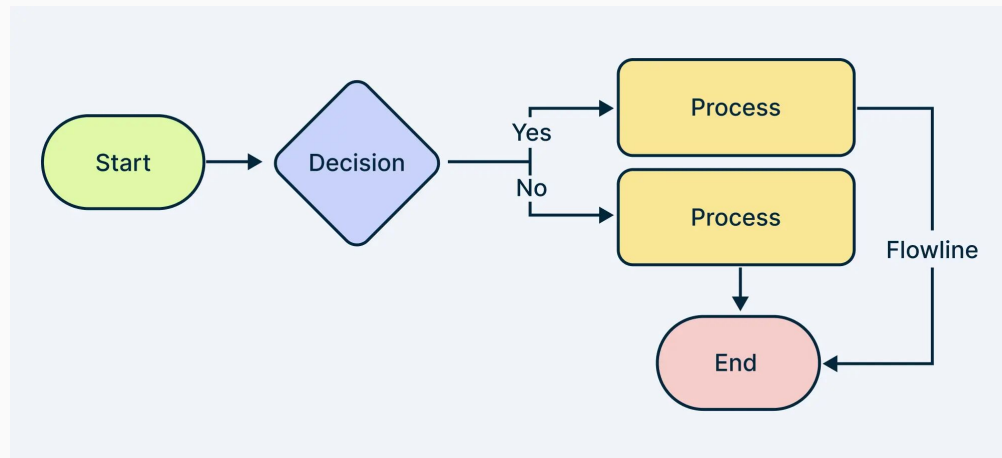
## نصب N8N با Docker

A screenshot of a web browser showing the n8n setup page. The browser's address bar shows 'localhost:5678/setup'. The page has a dark header with the n8n logo and the text 'n8n'. Below the header, there is a white box with the title 'Set up owner account'. Inside this box, there are four input fields labeled 'Email \*', 'First Name \*', 'Last Name \*', and 'Password \*'. Below the password field, there is a note: '8+ characters, at least 1 number and 1 capital letter'. At the bottom of the box, there is a checkbox labeled 'I want to receive security and product updates' and a red 'Next' button.

مرحله ۴ – ورود به محیط n8n ✓

حالا وارد پنل N8N میشیم:

## فلوچارت چیست؟



فلوچارت یعنی یک نقشه تصویری که با شکل‌های ساده نشون می‌ده یک کار قراره چطور و به چه ترتیبی انجام بشه

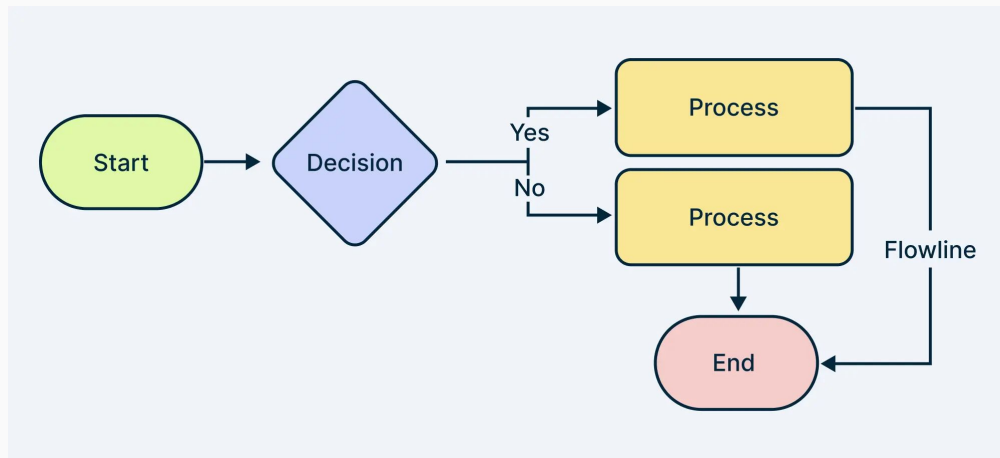
**مثال:**

اگر بارون میاد → چتر بردار  
اگر نه → بدون چتر برو بیرون.

---

با فلوچارت، شما یاد می‌گیرین چجوری یه کار رو به بخش‌های کوچک تقسیم کنید، و این پایه‌ی استفاده از n8n هست.

## Start در فلوچارت



### ۱. شروع (Start) ✓

♦ یعنی از کجا قراره کار ما شروع بشه.

📌 مثال واقعی:

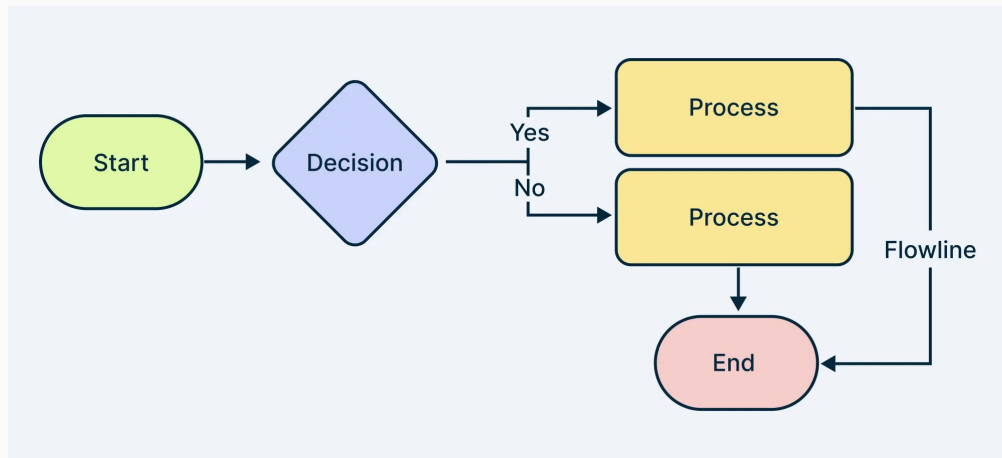
«هر روز صبح که بیدار می‌شی» → این همون شروع روزته!

📌 در n8n:

نودی مثل Manual Trigger یا Webhook کار رو شروع می‌کنه.

با فلوچارت، شما یاد می‌گیرین چجوری یه کار رو به بخش‌های کوچک تقسیم کنید، و این پایه‌ی استفاده از n8n هست.

## End در فلوچارت



### ۲. پایان (End) ✓

♦ یعنی کار کی تموم می‌شه و دیگه ادامه نداره.

📌 مثال واقعی:

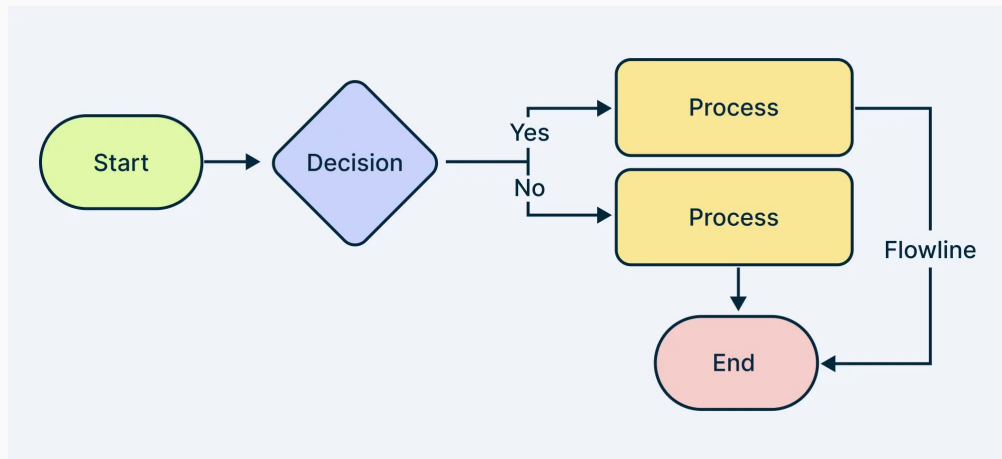
«بعد از اینکه مسواک زدی و صبحونه خوردی، می‌ری سرکار»  
→ اینجا برنامه صبحگاهی تموم شده.

📌 در n8n:

آخرین نود مثلاً ارسال Email یا ذخیره در Google Sheet هست که کار نهایی رو انجام می‌ده.

با فلوچارت، شما یاد می‌گیرین چجوری یه کار رو به بخش‌های کوچک تقسیم کنید، و این پایه‌ی استفاده از n8n هست.

## Process در فلوچارت



### ۳. عملیات (Process) ✓

♦ یعنی انجام یک کار مشخص، مثل محاسبه، ذخیره‌سازی، یا فرستادن یک پیام.

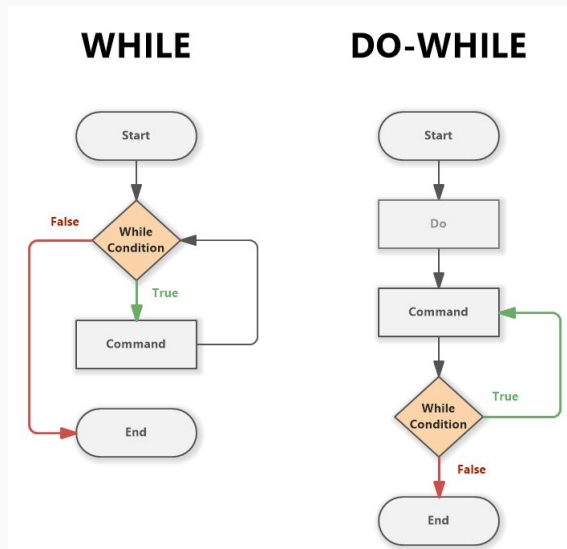
📌 مثال واقعی:

«بیدار شو → قهوه درست کن → قهوه رو بخور»  
هر کدام از این کارها به عملیات هستند.

📌 در n8n:

نودهایی مثل HTTP Request (دریافت اطلاعات)، Email (ارسال پیام) یا Set (ساخت داده) همین نقش رو دارن.

با فلوچارت، شما یاد می‌گیرین چجوری به کار رو به بخش‌های کوچک تقسیم کنید، و این پایه‌ی استفاده از n8n هست.



## ✓ ۶. حلقه (Loop)

♦ یعنی یک کاری رو تا زمانی که یه شرط خاص برقرار باشه، تکرار کن.

📌 مثال واقعی:

«لیست خرید داری → برای هر مورد در لیست → بررسی کن داری یا نه»

📌 در n8n:

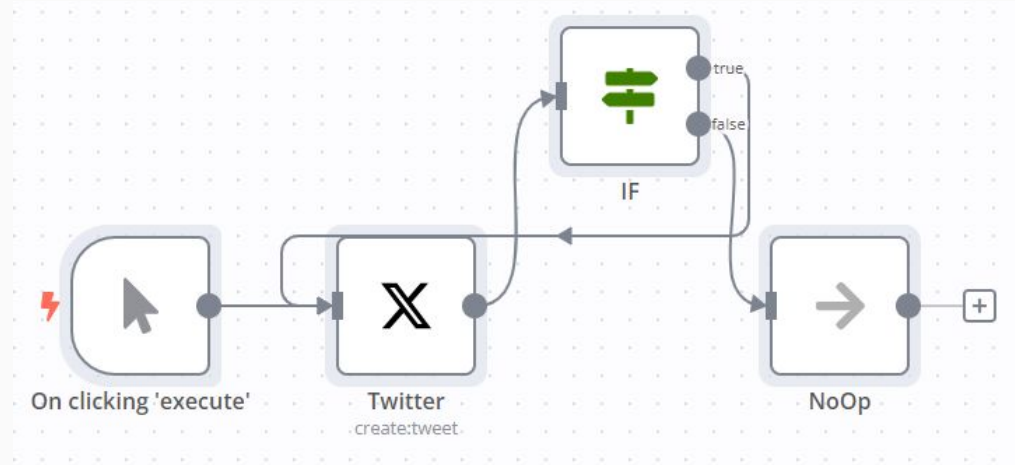
با Sub-workflow یا نودهایی مثل SplitInBatches می‌تونی لیستی از آیتم‌ها رو یکی‌یکی پردازش کنی.

با فلوچارت، شما یاد می‌گیرین چجوری یه کار رو به بخش‌های کوچک تقسیم کنید، و این پایه‌ی استفاده از n8n هست.




## Workflow یعنی یک برنامه ساده که به کامپیوتر می‌گی "اگه این اتفاق افتاد، این کارو بکن"

اگه یکی فرم ثبت‌نام پر کرد، اطلاعات رو ذخیره کن و براش پیام بفرست. با n8n این کارها رو با چند تا جعبه (نود) کنار هم می‌چینی، و اون مثل یه ربات از اول تا آخرش رو خودش انجام می‌ده. یعنی دیگه لازم نیست هر بار خودت دستی کاری انجام بدی؛ همه‌چی خودکار پیش می‌ره.





```
{ } names.json > ...  
1  {  
2    "name": "مینا",  
3    "age": 28,  
4    "phone": "09123456789"  
5  }  
6  
7  |
```

تصور کن یه جعبه داری که توش اطلاعات آدم‌ها رو گذاشتی: 

- اسمشون
- سنشون
- شماره‌شون

تعاریف:

- ♦ سمت چپ اسم هر چیزه (مثلاً name)،
- ♦ سمت راست، مقدار شه (مثلاً "مینا")



## JSON به زبان ساده

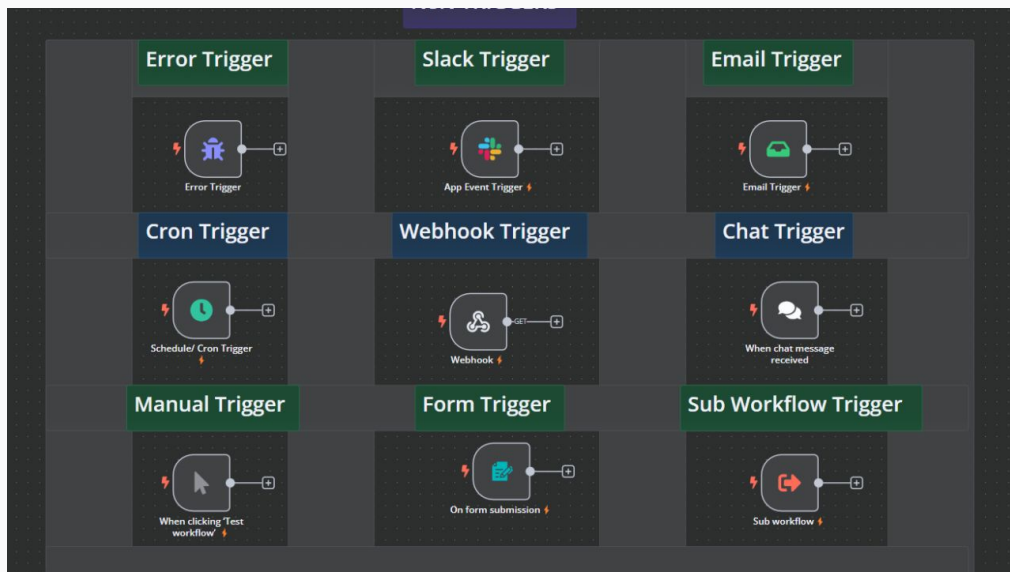


```
{ } names.json > { } 3 > abc name
```

```
1  [
2      {
3          "name": "مینا",
4          "age": 28,
5          "phone": "09123456789"
6      },
7      {
8          "name": "مجید",
9          "age": 5,
10         "phone": "09347634233"
11     },
12     {
13         "name": "سارا",
14         "age": 22,
15         "phone": "09123456788"
16     },
17     {
18         "name": "علی",
19         "age": 30,
20         "phone": "09234567890"
21     }
22 ]
```

آرایه (Array) توی JSON یعنی یه 🛒

لیست از چند چیز مشابه که کنار هم توی یه مجموعه نگه‌داری می‌شن.



## Trigger یعنی اون چیزی که باعث می‌شه جریان کار (Workflow) شروع بشه؛

مثل یه زنگ در که وقتی فشارش می‌دی، بقیه کارها راه می‌افته.

مثلاً می‌تونی بگی: “هر روز صبح ساعت ۸ کار رو شروع کن”، یا “هر وقت کسی فرم سایت رو پر کرد، شروع کن”.

این «شروع‌کننده‌ها» می‌شن Trigger. یعنی تا اتفاقی نیفته، n8n کاری نمی‌کنه؛ ولی به محض اینکه اون شرط اتفاق افتاد، بقیه کارها یکی یکی انجام می‌شن.



## If یعنی تصمیم‌گیری. درست مثل وقتی که توی زندگی روزمره می‌گی:

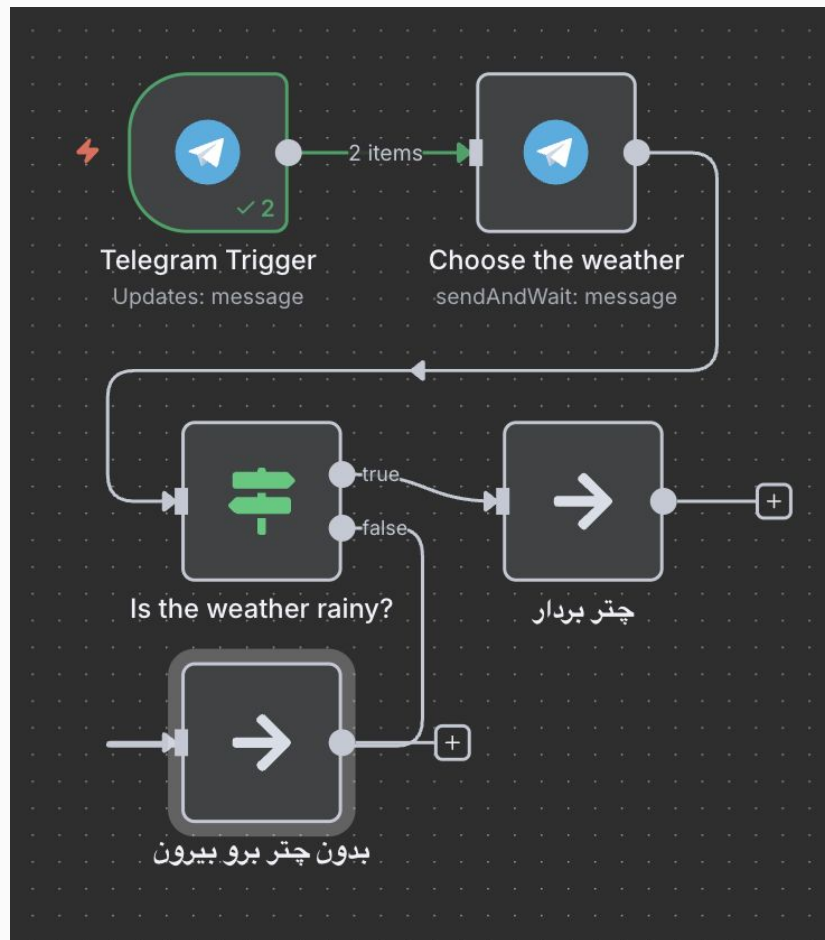
مثال ۱ - چتر برداشتن: 🌂

“اگر هوا بارونی بود → چتر بردار

وگرنه → بدون چتر برو بیرون”

→ توی n8n:

با استفاده از If بررسی می‌کنی که وضعیت هوا چطور، بعد تصمیم می‌گیری پیام هشدار بدی یا نه.





## Switch کار مقایسه و تصمیم‌گیری رو ساده می‌کنه!

تصور کن توی یک مسیر داری حرکت می‌کنی و به یک چهارراه می‌رسی.

توی این چهارراه، باید بر اساس یک شرط تصمیم بگیری کدوم مسیر رو ادامه بدی:

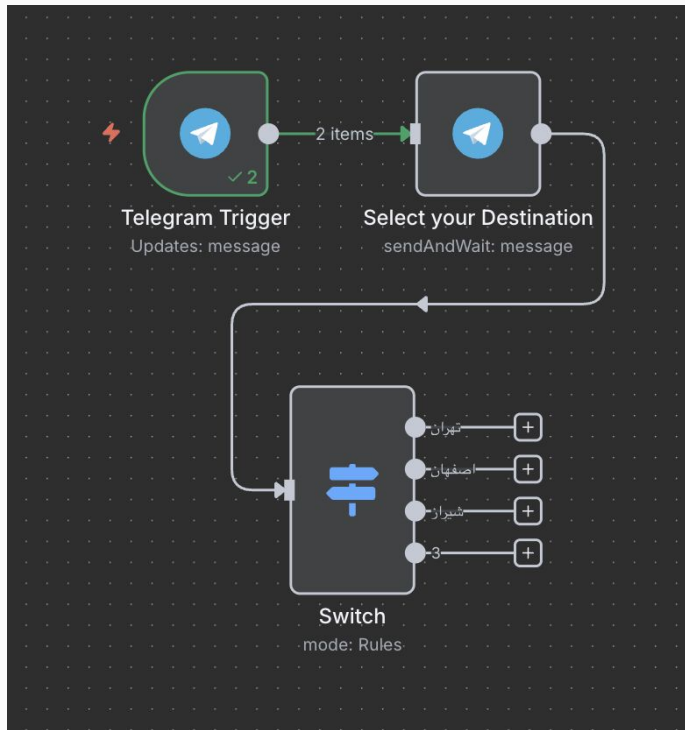
اگر شهر "تهران" بود → ایمیل به تیم تهران ارسال بشه

اگر شهر "اصفهان" بود → ایمیل به تیم اصفهان ارسال بشه

اگر هیچکدوم نبود → ایمیل به تیم اصلی بره

چه زمانی از Switch به جای IF استفاده کنیم؟

- نیازی به چند تا IF پشت سر هم نیست
- همه شرط‌ها رو یکجا مدیریت می‌کنید





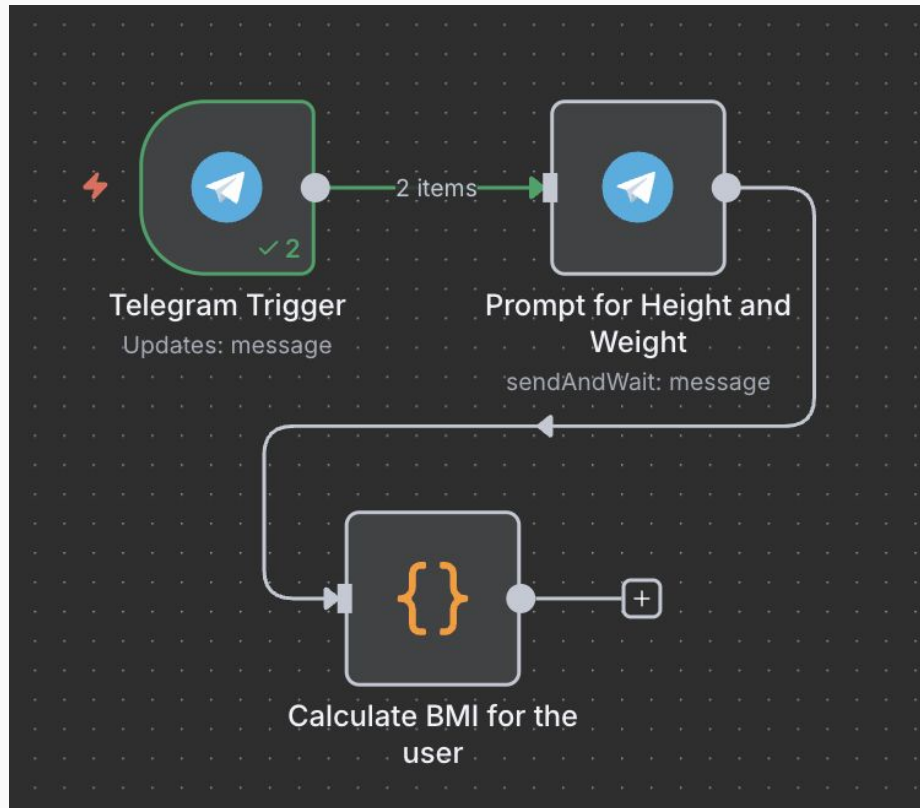
## با Code می‌تونن یک قطعه کد JavaScript بنویسن تا دقیقاً همون کاری که می‌خواه رو انجام بده.

در n8n بیشتر کارها با **Node های آماده** انجام می‌شه (مثل ارسال ایمیل، گرفتن دیتا از API یا فیلتر کردن داده‌ها).

اما گاهی وقت‌ها کاری که می‌خواه انجام بدی آماده وجود نداره یا می‌خواه یک تغییر خیلی خاص روی داده انجام بدی.

اینجاست که **Code Node** به کمکت میاد.

با Code Node می‌تونن یک قطعه کد JavaScript بنویسن تا دقیقاً همون کاری که می‌خواه رو انجام بده.





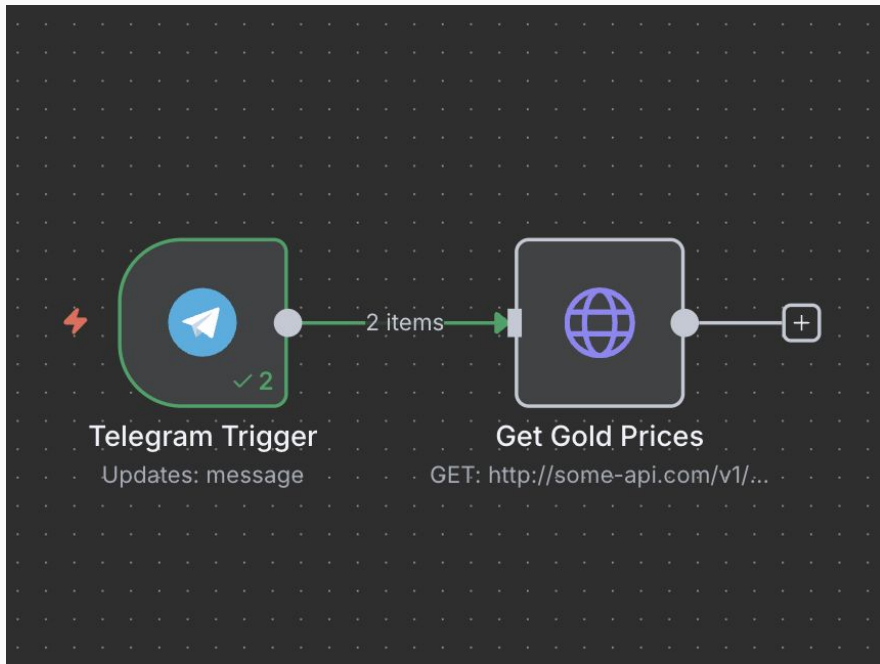
## با HTTP Request میتونی از آدرس های مختلف اطلاعات بگیری یا اطلاعاتی رو بارگزاری کنی!

تصور کن n8n یک آدم پیام‌رسانه که می‌تونه بره در یک  
سایت یا سرویس اینترنتی رو بزنه و بگه:

«سلام، من این اطلاعات رو می‌خوام» یا «سلام، این  
اطلاعات رو برات فرستادم»

این کار همون HTTP Request هست.

یعنی با این n8n، Node می‌تونه با هر سرویسی که API  
داره حرف بزنه و اطلاعات بگیره یا بفرسته.





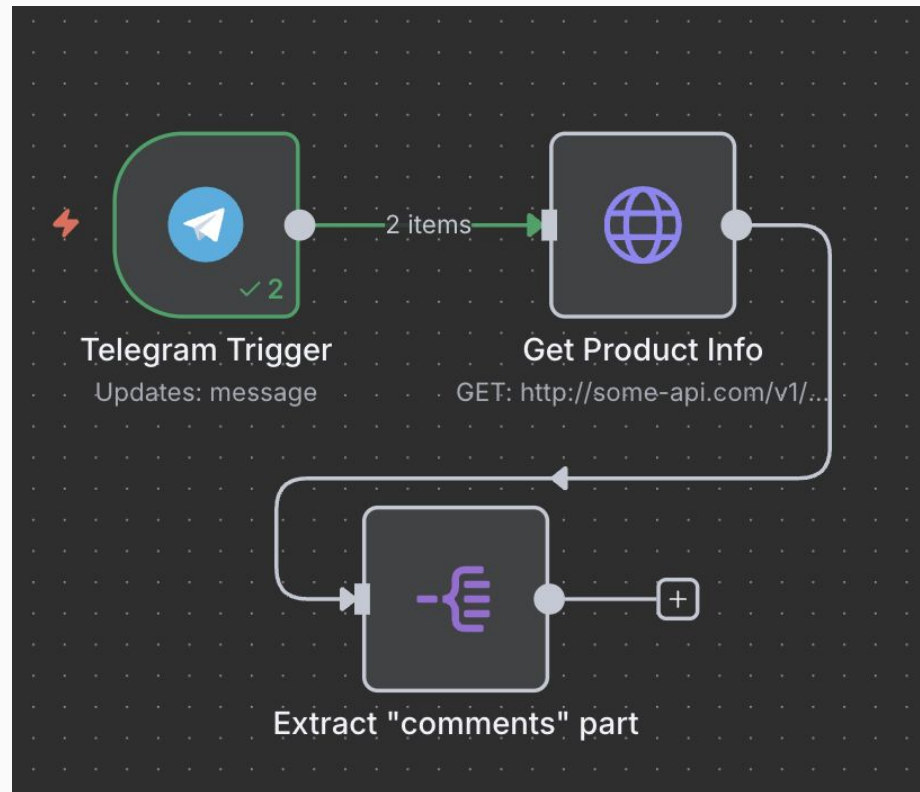
## با Split Out میتونی از داخل یک بسته بزرگ، بسته های کوچک رو بیرون بکشی!

فرض کن یه بسته بزرگ داری که داخلش چند تا جعبه کوچیکه.

وقتی این بسته به دستت میرسه، اگه بخوای روی هر جعبه جداگانه کاری انجام بدی، باید اول اون رو تکتک باز کنی.

### Split Out در n8n دقیقاً همین کار رو می‌کنه:

- وقتی یک Node چند آیتم (Item) تولید کرده،
- Split Out اون‌ها رو یکی‌یکی جدا می‌کنه
- بعد هر کدوم رو جداگانه می‌فرسته برای مرحله بعدی



# مفاهیم کلیدی اولیه در AI



## Frequency penalty مفهوم



### مثال روزمره

فرض کن از یه هوش مصنوعی می‌خوای درباره قهوه متن بنویسه.

**Frequency Penalty = 0 (هیچ جریمه‌ای نیست)**

«قهوه خیلی خوبه. قهوه انرژی میده. قهوه صبح‌ها بهترین چیزه. قهوه واقعا عالیه.»  
(می‌بینی؟ هی قهوه رو تکرار کرده)

**Frequency Penalty = 1 (مقداری جریمه)**

«قهوه خیلی خوبه. این نوشیدنی انرژی میده. صبح‌ها بهترین انتخابه.»  
(این بار سعی کرده مترادف بپاره و کمتر تکرار کنه)

**Frequency Penalty = 2 (خیلی زیاد)**

«این نوشیدنی صبحگاهی شبیه جادو عمل می‌کنه. بوی معطرش مغز رو بیدار می‌کنه و حس سرزندگی میده.»  
(اینجا دیگه کلمه‌ی «قهوه» رو کمتر یا حتی اصلاً نگفت، چون جریمه خیلی بالاست)

### توضیح مفهوم Frequency Penalty

تصور کن یه بچه کوچیک داری که وقتی یه کلمه رو یاد می‌گیره، همش تکرارش می‌کنه.

مثلاً هی می‌گه:

«مامان، مامان، مامان، مامان...»

مدل‌های هوش مصنوعی هم بعضی وقت‌ها همین‌طور می‌شن. یه کلمه یا عبارت رو زیاد تکرار می‌کنن.

Frequency Penalty مثل یه قانون یا «جریمه» است که می‌گه:

«هر چی بیشتر یه کلمه رو تکرار کنی، کمتر اجازه دارم دوباره بگم!»

## مفهوم Max Number of Tokens



### توکن دقیقاً چیه؟ 🔍

هر کلمه یا تکه کلمه یه توکن حساب میشه.

(تقریباً هر ۱۰۰ توکن معادل ۷۵ کلمه ی انگلیسیه. در فارسی کمی بیشتر میشه چون کلمات طولانی ترن.)

### مثال روزمره 📌

از مدل می‌خوای درباره قهوه توضیح بده.

Maximum Tokens = 20

«قهوه یک نوشیدنی محبوب است که صبح‌ها انرژی می‌دهد.»  
(خیلی کوتاه، چون سقف کم گذاشتی.)

### توضیح مفهوم Max Number of Tokens: 📌

تصور کن داری به یه بچه می‌گی:

«یه قصه برام بگو.»

اگه بهش بگی فقط ۲ جمله بگو → قصه کوتاه و جمع‌وجور میشه.

اگه بگی می‌تونم ۲۰ جمله بگی → قصه مفصل‌تر میشه.

توکن‌ها هم توی هوش مصنوعی مثل همون تعداد جمله‌ها یا کلمه‌ها هستن.

یعنی Maximum Number of Tokens = حداکثر طول جوابی که مدل اجازه داره بده.

## مفهوم Presence Penalty



از مدل می‌خواهی یه لیست از سرگرمی‌ها بده.

**Presence Penalty = 0**

کتاب خواندن، فیلم دیدن، کتاب داستان، تماشای سریال  
(همون ایده‌های تکراری رو میاره چون مشکلی با تکرار نداره)

**Presence Penalty = 1**

کتاب خواندن، فیلم دیدن، آشپزی، کوه‌نوردی  
(سعی کرده بیشتر تنوع بیاره)

**Presence Penalty = 2**

کوه‌نوردی، موسیقی، نقاشی، باغبانی  
(دیگه سراغ کتاب و فیلم نرفته چون خیلی جریمه داره)

Presence Penalty به زبان ساده 🎯

تصور کن یه دوست داری که وقتی ازش می‌پرسی: «سرگرمی‌هات چیه؟»

همیشه میگه:

«کتاب خوندن، فیلم دیدن، کتاب خوندن، فیلم دیدن...»

خب این کلافه‌کننده میشه

Presence Penalty مثل یه قانونیه که به هوش مصنوعی میگه:

«اگه یه چیزی رو

قبلاً گفتی، دوباره خیلی راحت سراغش نرو. بهتره چیزای جدید بیاری.»

## مفهوم Temperature



سؤال: «یه جمله درباره قهوه بگو»

Temperature = 0.1

«قهوه یک نوشیدنی محبوب است.»  
(خیلی خشک و مطمئن)

Temperature = 0.7

«قهوه صبح‌ها مثل یک جادو عمل می‌کند و آدم را سرحال می‌آورد.»  
(خلاق‌تر و طبیعی‌تر)

Temperature = 1.5

«قهوه سفری به دنیای عطر و طعم است که قلبت را بیدار می‌کند و ذهنت را  
به پرواز درمی‌آورد.»  
(خیلی شاعرانه و غیرقابل‌پیش‌بینی)

Sampling Temperature به زبان ساده 

تصور کن می‌خوای یه داستان تعریف کنی.

- اگه خیلی جدی و منطقی باشی → همیشه داستان رو شبیه هم میگی.
- اگه یه کم آزادتر باشی → هر بار داستان یه twist جدید داره.
- اگه خیلی هیجانی باشی → هر دفعه یه داستان عجیب‌غریب می‌گی!

Sampling Temperature هم همین‌طوره:

- عدد پایین (۰ تا ۰.۳) → جواب دقیق، منطقی و قابل‌پیش‌بینی.
- عدد متوسط (۰.۵ تا ۰.۷) → جواب متنوع‌تر ولی هنوز منطقی.
- عدد بالا (۱ تا ۲) → جواب خیلی خلاق، غیرقابل‌پیش‌بینی، گاهی حتی عجیب.

# کاربرد Node های مختلف AI در N8N

---

The N8N



## 🎯 معایب Basic LLM Chain

### خیلی ساده و محدود

فقط یه ورودی می‌گیره و جواب می‌ده.

یعنی نمی‌تونه چند مرحله فکر کنه یا کارهای پیچیده (مثل بازیابی اطلاعات از دیتابیس یا چند بار سؤال پرسیدن) انجام بده.

### عدم مدیریت حافظه (Memory)

- مدل جواب سؤال فعلی رو می‌ده و هیچ یادی از سؤال قبلی نداره.
- یعنی نمی‌تونه مکالمه ادامه‌دار داشته باشه.

مثال: 📌

- تو: «ا کشور تولیدکننده قهوه رو بگو.»
- مدل: «برزیل، ویتنام، کلمبیا...»
- تو: «سومی کدوم بود؟»
- 🍌 مدل دیگه یادش نمیداد "کلمبیا" سوم بوده.

## 🎯 Basic LLM Chain یعنی چی؟

وقتی با یک مدل زبانی (LLM) کار می‌کنیم، معمولاً کارمون اینه که:

1. یه سؤال یا ورودی به مدل بدیم.

2. یه خروجی از مدل بگیریم.

اما توی عمل همیشه این‌قدر ساده نیست.

ما معمولاً قبل از اینکه متن کاربر رو به مدل بدیم، باید یک سری کار انجام بدیم:

- ورودی رو درست قالب‌بندی کنیم (Prompt ساختن)
- شاید اطلاعات اضافی (مثلاً از دیتابیس یا API) بهش اضافه کنیم
- بعد جواب مدل رو بگیریم و پردازش کنیم

به این جریان (Pipeline) می‌گن Chain.



### ⚡ کاربردهای واقعی در n8n

1. تحلیل نظرات کاربران → استخراج نام محصول + امتیاز + احساس (مثبت/منفی).
2. پردازش ایمیل‌ها → جدا کردن نام مشتری، شماره سفارش، مشکل مطرح‌شده.
3. ورود اطلاعات اتوماتیک → گرفتن داده از متن و ذخیره مستقیم در Google Sheet یا CRM.
4. استخراج داده از مقاله‌ها یا گزارش‌ها → مثل اسم کشورها، تاریخ‌ها یا آمارهای مهم.

### 🎯 Information Extractor در n8n چیه؟

این به کامپوننت AI هست که وقتی به مدل متنی (مثل GPT) وصل میشه، کمک می‌کنه از یه متن طولانی، فقط اطلاعات کلیدی و ساختارمند رو بیرون بکشی.

یعنی به جای اینکه مدل فقط متن آزاد بده، خروجی‌اش همیشه یه ساختار JSON یا جدول که راحت بتونی توی workflow استفاده کنی.

خیلی وقت‌ها توی پروژه‌های اتوماسیون، متن ورودی ما شلخته‌ست:

- ایمیل مشتری
- نظر کاربر روی محصول
- متن مقاله یا گزارش
- چت با پشتیبانی

ولی ما دنبال چیز خاصی هستیم مثل: اسم، شماره، تاریخ، قیمت، موضوع، احساس (مثبت/منفی)



### ⚡ کاربردهای واقعی در n8n

چت بات پرسش و پاسخ → جواب ساده و مستقیم بده.

FAQ اتوماتیک برای سایت → سوال کاربر رو بگیره و از دیتای آماده جواب بسازه.

کمک آموزشی → کاربر سؤال علمی بپرسه و جواب بگیره.

پشتیبانی مشتری → سوال کاربر رو مستقیم جواب بده (بدون نیاز به حافظه طولانی).

### 🎯 Question and Answer Chain در n8n چیه؟

یک Chain مخصوص برای پاسخ به سؤال هاست.

کارش اینه که:

1. سوال کاربر رو می گیره.
2. اون رو با یک Prompt Template ترکیب می کنه.
3. می ده به LLM (مدل هوش مصنوعی).
4. جواب برمی گردونه.

خیلی شبیه Basic LLM Chain هست، ولی تمرکزش روی پرسش و پاسخ مستقیمه.





### کاربردهای واقعی در n8n ⚡

#### 1. تحلیل نظرات مشتری‌ها

گرفتن نظرها از دیجی‌کالا یا گوگل ریویو → تبدیل به مثبت/منفی/خنثی → ذخیره توی Dashboard یا Google Sheet.

#### 2. پشتیبانی مشتری

وقتی کاربر ایمیل می‌زنه، سیستم اتوماتیک تشخیص می‌ده متن عصبیه یا خنثیه → درخواست فوری به تیم پشتیبانی ارسال بشه.

#### 3. شبکه‌های اجتماعی

• جمع‌آوری توییت‌ها یا کامنت‌ها → تحلیل احساسات → متوجه بشی مردم در کل خوشحال، ناراحت یا بی‌تفاوت هستن نسبت به برند یا محصولت.

#### 4. مارکتینگ

• قبل و بعد از کمپین تبلیغاتی → تحلیل کنی نظرها چقدر مثبت‌تر یا منفی‌تر شدن.

### Sentiment Analysis در n8n چیه؟ 🎯

تحلیل احساسات (Sentiment Analysis) یعنی یک متن رو بخونی و بفهمی نویسنده اون متن چه حسی داشته:

- مثبت 😊
- منفی 😞
- یا خنثی 😐

مدل‌های هوش مصنوعی می‌تونن از روی متن، لحن، و انتخاب کلمات، احساس پشت اون رو تشخیص بدن.



### ⚡ کاربردهای واقعی در n8n

#### 1. مدیریت ایمیل‌ها

ایمیل مشتری رو بگيره → دسته‌بندی کنه به "پشتیبانی فنی"، "سفارش"، "مالی".  
بعد خودکار به تیم مربوطه ارجاع بده.

#### 2. تحلیل شبکه‌های اجتماعی

توییت‌ها یا کامنت‌ها رو دسته‌بندی کنه به "محصولات"، "خدمات"، "شکایت‌ها"،  
"پیشنهاده‌ها".

#### 3. سازماندهی محتوای متنی

متن‌های خبری رو بگيره → دسته‌بندی کنه به "ورزشی"، "اقتصادی"، "سیاسی".

#### 4. چت‌بات‌ها

• کاربر چیزی می‌نویسه → سیستم تشخیص می‌ده سوالش درباره "پرداخت"، "ارسال" یا  
"محصول" بوده → جواب مناسب بده.

### 🎯 Text Classifier در n8n چیه؟

Text Classification (طبقه‌بندی متن) یعنی یه متن رو بگیری و مشخص کنی که توی  
کدوم دسته یا کلاس قرار می‌گیره.

یعنی به جای اینکه فقط احساس متن رو (مثبت/منفی) بفهمیم (مثل Sentiment  
Analysis)، این بار می‌خوایم متن رو بر اساس موضوع، برچسب یا گروه دسته‌بندی کنیم.