

خانه فناوری تهران

www.Fanavari.co



# الگوریتم چیست؟

(آشنایی با مبانی برنامه نویسی)

جلسات روز ۱ دوره - بخش ۱

## در این بخش چه می‌آموزیم؟

تعریف و قوانین  
(شروع، پایان، ترتیب)



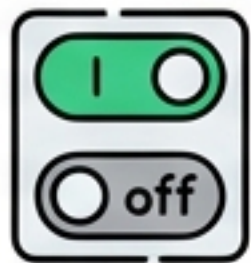
مثال ربات و اتوبوس  
(درک منطق ماشین)



الگوریتم‌های زندگی  
(چای و نیمرو)



برنامه‌نویسی استاتیک  
و داینامیک





## الگوریتم: ریتم و الگوی حل مسئله

کلمه الگوریتم از دو بخش «الگو» و «ریتم» تشکیل شده است. برنامه‌نویسی یعنی پیدا کردن ریتم و الگوی اجرای یک کار.

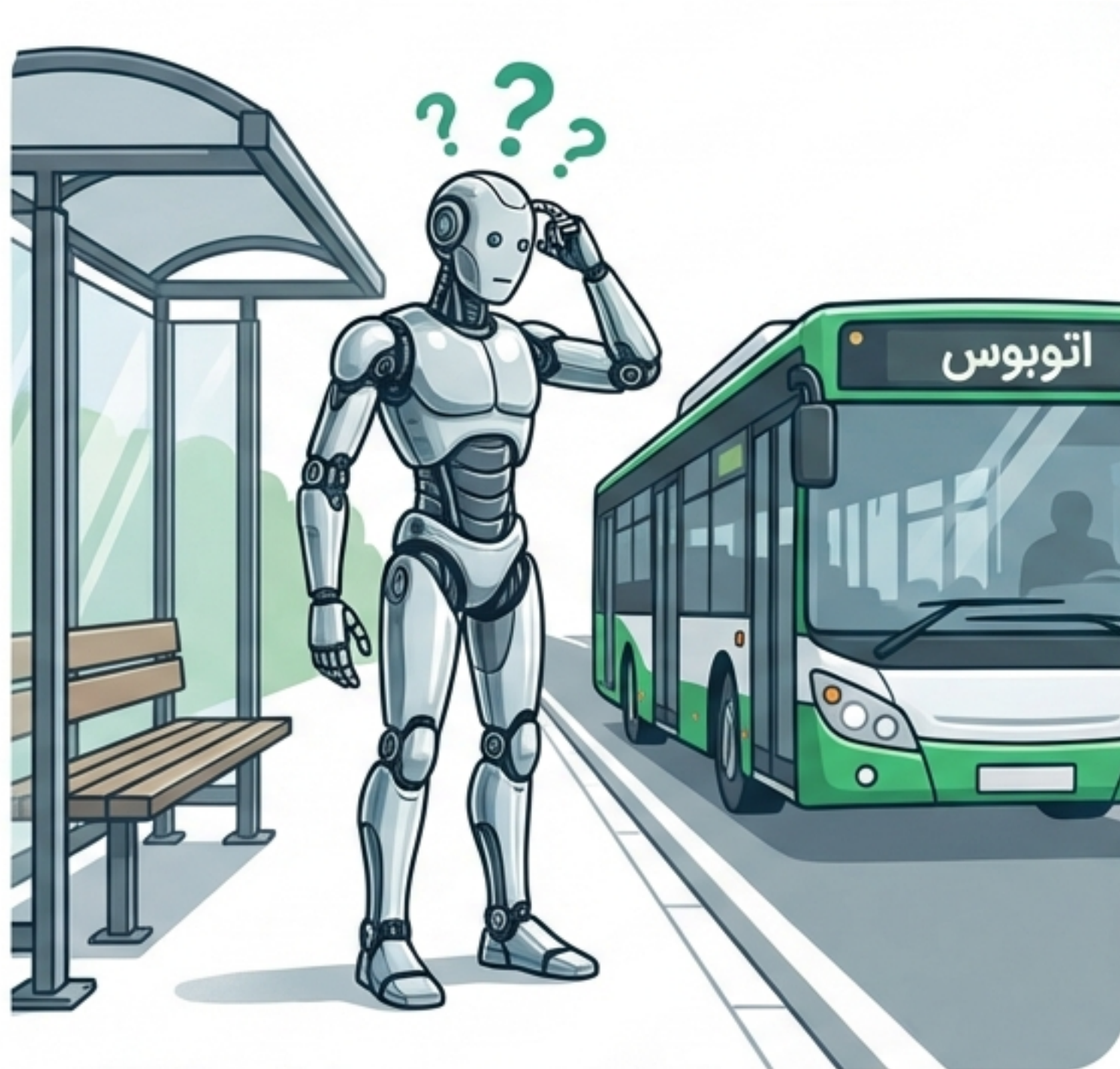
مهم نیست چقدر به زبان‌های پایتون یا جاوا مسلط باشید، یا حتی از هوش مصنوعی (ChatGPT) کمک بگیرید؛ اگر «منطق» و «الگوریتم» حل مسئله را ندانید، هیچ کدی کار نخواهد کرد.

**الگوریتم، مغز متفکر کد شماست.**

## سه ویژگی حیاتی یک الگوریتم

Start → Steps → End

1. **دارای شروع و پایان باشد:** هر برنامه‌ای باید نقطه‌ی آغاز و پایان مشخص داشته باشد (Start & End).
  2. **مراحل مشخص باشد:** گام‌ها باید دقیق و بدون ابهام باشند (Steps).
  3. **جزئیات کافی باشد:** ماشین باید دقیقاً بداند چه کاری انجام دهد (Details).
- اگر این سه مورد رعایت نشود، برنامه یا اجرا نمی‌شود یا در یک چرخه بی‌پایان گیر می‌کند.



## مثال اول: ربات و اتوبوس

تصور کنید یک ربات پیشرفته داریم. او راه رفتن و دیدن را بلد است (سخت‌افزار سالم)، اما نمی‌داند چگونه سوار اتوبوس شود.

### مشکل کجاست؟

اگر به او بگوییم «سوار شو»، او فقط در ایستگاه می‌ایستد. اتوبوس می‌آید و می‌رود و ربات حرکتی نمی‌کند. چرا؟

چون الگوریتم (دستورالعمل) سوار شدن را به او نداده‌ایم. هوش مکانیکی بدون دستورالعمل منطقی، بی‌استفاده است.

## رفع باگ: نوشتن دستورالعمل دقیق

۱. شروع
۲. در ایستگاه بنشین و منتظر باش.
۳. **شرط مهم:** وقتی اتوبوس رسید و **ایستاد (Stop)**، بلند شو.
۴. سوار شو و روی صندلی بنشین.
۵. در مقصد پیاده شو.
۶. پایان

**درس مهم (Bug Fix):** اگر نگویم «منتظر باش تا اتوبوس بایستد»، ممکن است ربات جلوی اتوبوس در حال حرکت بپرد! باگ‌های برنامه‌نویسی معمولاً ناشی از همین جزئیات فراموش شده هستند.



## تمرین کلاسی: الگوریتم درست کردن چای

هیچ دو برنامه‌نویسی دقیقاً مثل هم فکر نمی‌کنند.

- روش اول (ساده): استفاده از چای کیسه‌ای (Lipton) برای یک نفر. (سریع و ساده)

- روش دوم (دقیق): چک کردن آب کتری، دم کردن چای، سرو برای چند نفر. (دقیق و کامل)

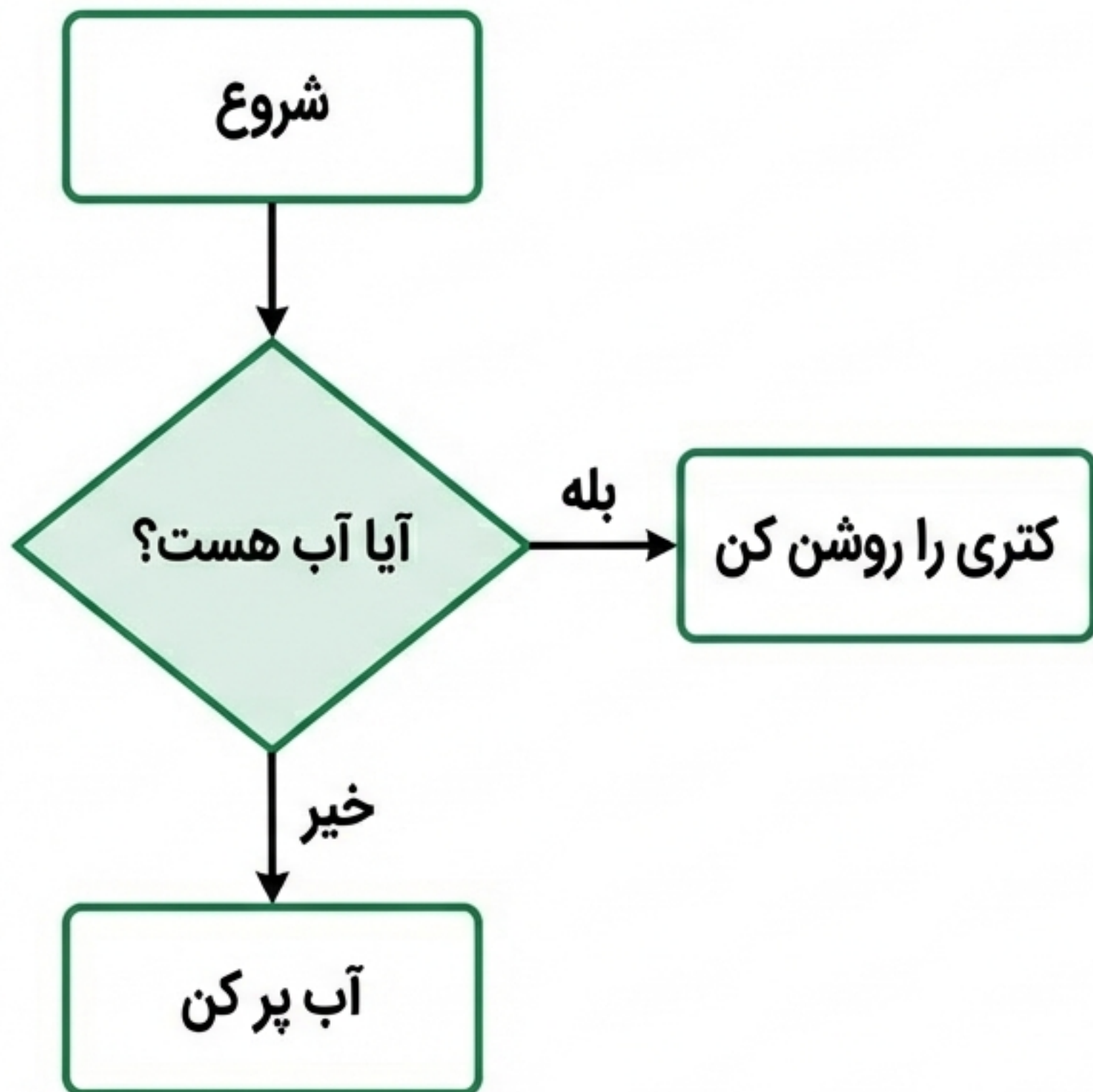
هر دو روش درست هستند، اما «رویکرد» و «نیاز» کاربر تعیین می‌کند کدام بهتر است.

## مدیریت خطا (Error Handling) و کاهش ریسک

**سناریو:** یکی از دانشجویان مرحله‌ای اضافه کرد: «اول چک کن بین کتری آب دارد یا نه؟»

● **چرا مهم است:** این جزئیات اضافه نیست؛ این کاهش ضریب خطا است.

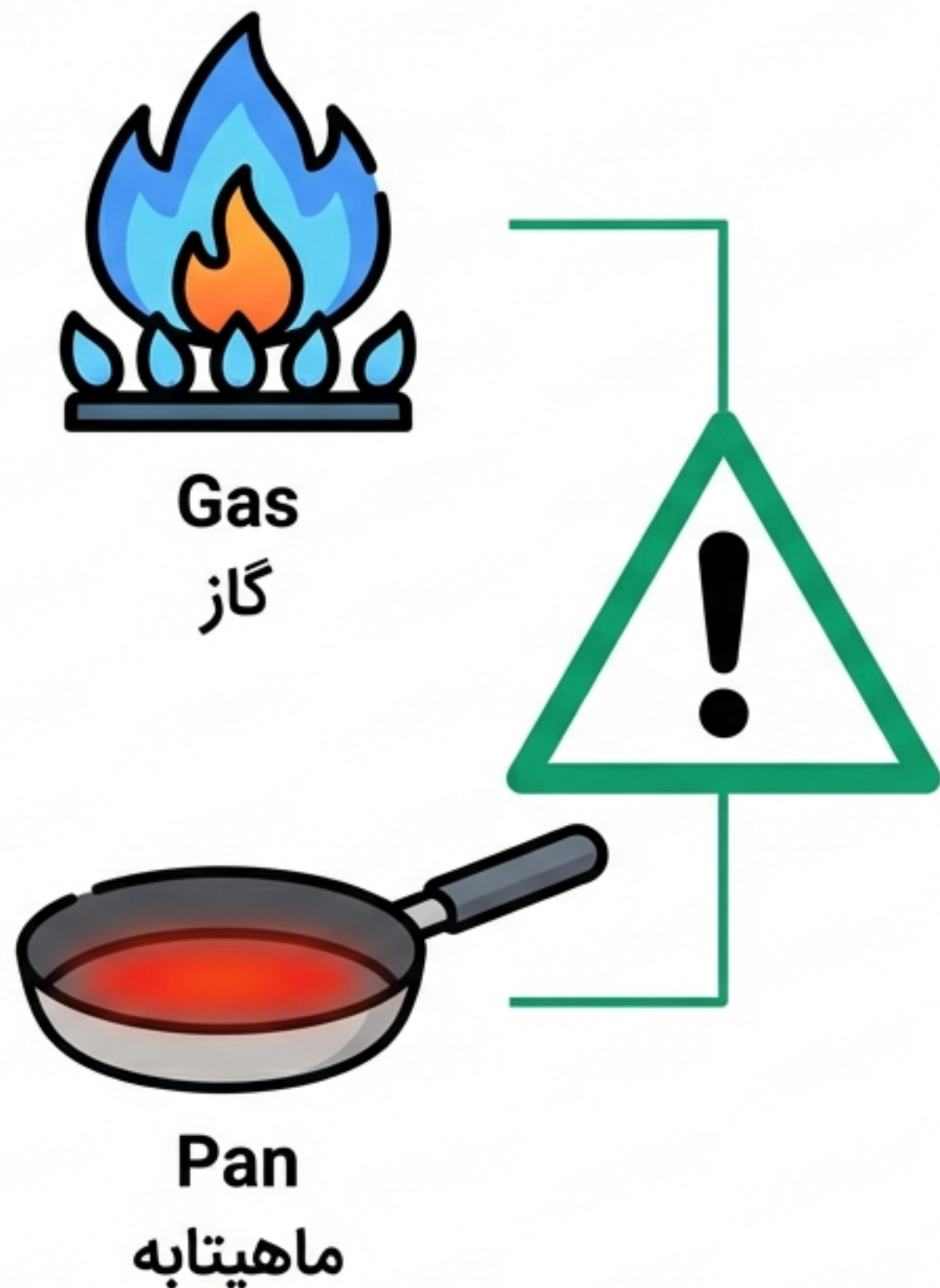
**درس مهم:** یک برنامه خوب همیشه حالت‌های غیرعادی (مثل نبودن آب) را پیش‌بینی می‌کند تا پیش‌بینی می‌کند. تا برنامه متوقف نشود (Crash نکند).



## تمرین دوم: نیمرو و خطر «حلقه بی‌نهایت»



- در دستور پخت گفتیم: «تخم‌مرغ را هم بزن.»  
اما نگفتیم چقدر!
- ربات یا کامپیوتر تا ابد به هم زدن ادامه می‌دهد.
- به این وضعیت **Infinite Loop** (حلقه بی‌نهایت) می‌گویند.
- همیشه باید شرط پایان را ذکر کنید: «۵ دقیقه صبر کن» یا «تا زمانی که پخته شود».



## کامپیوترها حرف شنو اما نادان هستند!

**\*\*The Bug\*\***

یک دانشجو نوشت: «صبر کن تا گاز داغ شود.»

**\*\*The Reality\*\***

گاز داغ نمی‌شود، **ماهیتابه** داغ می‌شود!

● **درس: Lesson\*\***

کامپیوتر منظور شما را حدس نمی‌زند. اگر دستور اشتباه بدهید، عیناً همان را اجرا می‌کند و به خطا (Error) می‌خورد. دقت در کلمات حیاتی است.

## ترتیب و تفکیک مراحل

- «روغن را بریز و گاز را روشن کن.»
- آیا این دو کار همزمان انجام می‌شوند؟ خیر.

● بهترین آمیر می‌شود:

۱. گاز را روشن کن.
۲. روغن را داخل تابه بریز.

- ترکیب کردن کارهای نامرتب (مثل شکستن تخم‌مرغ و روشن کردن گاز در یک خط) باعث پیچیدگی و باگ می‌شود.

Turn on Gas

Pour Oil



## ورودی‌ها (Inputs) و متغیرها

● The Issue: اگر بنویسیم «۲ تخم‌مرغ بشکن»، این برنامه فقط **فقط** برای ۱ نفر کار می‌کند. اگر **۱۵ نفر** باشند چه؟

● The Solution **اول پرس:** «تعداد نفرات چند تاست؟» (Input).

● Algorithm Update

1. شروع
2. تعداد نفرات را بشمار.
3. به ازای هر نفر، ۲ تخم‌مرغ بردار.

برنامه شما اکنون هوشمند و **مقیاس‌پذیر (Scalable)** است.

چند نفر؟

15

ثبت

## Static (ثابت)



منوی رستوران  
(غیر قابل تغییر)

## Dynamic (پویا)



تنظیمات شخصی  
(قابل تغییر)

## برنامه استاتیک (Static) در برابر داینامیک (Dynamic)

**\*\*Static (ثابت):**

مثل یک منوی رستوران که تغییر نمی‌کند.  
«نمک و فلفل را اضافه کن.» (کاربر حق انتخاب ندارد).

**\*\*Dynamic (پویا):**

به کاربر حق انتخاب می‌دهد.  
«آیا نمک می‌خواهی؟»

**\*\*Real World Example\*\***

- سایت دیجی‌کالا (قالب ثابت/استاتیک).
- داشبورد تنظیمات شخصی (تم تاریک/روشن/داینامیک).



## نکته طلایی: فراتر از وظیفه (The Extra Mile)

- **داستان:** استاد در پروژه دانشگاهی، علاوه بر لیست کردن ساکنین ساختمان (وظیفه اصلی)، امکان خروجی اکسل و گرافیک بهتر را هم اضافه کرد.
- **درس:** اگر کارفرما از شما  $X$  را می‌خواهد، شما  $Y + Y$  را تحویل دهید.
- **نتیجه:** در مصاحبه‌های استخدامی، این کارهای کوچک و خلاقانه (مثل برداشتن تابه از روی گاز در پایان کار، حتی اگر گفته نشده باشد) شما را از بقیه متمایز می‌کند.

## جمع‌بندی جلسه اول

- الگوریتم یعنی نوشتن دقیق مراحل (با کاغذ و خودکار) قبل از کدنویسی. برکه نشستا و خوواتی را بدارِ ا بیان قبل از کدنویسی.
- شروع، پایان، و جزئیات دقیق الزامی است.
- همیشه به فکر «شرط پایان» و «ورودی‌های کاربر» باشید.

در جلسه بعد، این منطق را تبدیل به کدهای واقعی خواهیم کرد.



خانه فناوری تهران

[www.Fanavari.co](http://www.Fanavari.co)



خانه فناوری تهران

پایان بخش اول

[www.Fanavari.co](http://www.Fanavari.co)