

[Open in app](#)

[Sign up](#)

[Sign in](#)

Medium



Search



Write



# What Is Data? Structured vs. Unstructured Data (Explained Simply for Beginners)



Wuraola Ademola-Shanu

[Follow](#)

17 min read · Aug 15, 2025



1



# WHAT IS DATA?

## STRUCTURED VS. UNSTRUCTURED DATA



**STRUCTURED**

VS.



**UNSTRUCTURED**

**EXPLAINED SIMPLY FOR BEGINNERS**

When I first heard the word “*data*,” I imagined rows of cold, lifeless numbers that only highly trained computer scientists could make sense of. In my mind, data was this untouchable, intimidating thing floating somewhere in complex databases, surrounded by code I’d never understand. If you’ve ever felt that way, you’re not alone.

But here’s the truth: data is everywhere, and you’ve been interacting with it for years without realising it. From the moment you check your phone in the morning to the moment you fall asleep at night, you’re creating and consuming data constantly. The way it’s stored, labelled, and organised is what determines whether it’s called “structured,” “unstructured,” or “semi-structured.”

In this guide, I’m going to break it all down in a way that makes it click... without jargon, without assumptions that you already “speak tech,” and without overwhelming you. By the time we’re done, you won’t just *understand* these terms; you’ll be able to spot them in your everyday life, explain them to someone else, and feel confident stepping into conversations about data without second-guessing yourself.

## TL;DR

- **Structured data** = neatly organised, easy to search (like a spreadsheet).
- **Unstructured data** = messy, free-form, doesn’t follow a set layout (like photos, videos, or handwritten notes).
- **Semi-structured data** = somewhere in between, has labels but isn’t rigid (like a JSON file or email headers).

**Why it matters:** Different types of data require different tools, storage systems, and analytical approaches. Once you can identify these types, working with data becomes far less intimidating and far more exciting.

## Introduction

This morning, before you ate breakfast or even finished your first cup of coffee, you created more data than you probably realise.

When you unlocked your phone, it recorded the time (**structured**). When you texted your friend a photo of your breakfast, that was an image file (**unstructured**). And when your food delivery app sent you a receipt, that was a mix of order details in a neat table and a note from the driver (**semi-structured**).

The thing is, data isn't some mysterious thing that only tech people understand. It's just... facts and information, stored in different ways. Understanding the type of data you have isn't just a technical skill, but the first step to making sense of the world's information. The way data is organised determines how easy it is to find, use, and learn from it.

That's what this guide is here to teach you. We'll go deep, not just into what structured and unstructured data mean, but why they matter, how they're stored, and how you can work with them — all with examples you already live with every day.

## What is data?

In its simplest form, **data is information**. It's the raw material from which knowledge is built: the facts, figures, measurements, observations, or descriptions that we collect from the world around us.

Data can be numbers, words, images, audio, sensor readings, clicks on a website, or even the temperature outside. What makes it “data” is not the format, but the fact that it’s recorded and stored in a way that can be referenced, analysed, and used to make decisions.

Think of data as the foundation of decision-making in every field. For example:

- Businesses use sales data to understand their customers and improve products.
- Scientists use experimental data to confirm or reject hypotheses.
- Governments use census data to allocate resources and plan infrastructure.
- Even you might use personal data like steps tracked on a smartwatch to make lifestyle choices like deciding whether you’ve earned dessert tonight. 😊

But here’s a key point: data on its own is just a collection of facts. The power of data comes when it is processed, organised, and interpreted. This is where the concepts of data engineering, data science, and analytics come in: *turning raw information into insights that can guide action.*

A crucial detail is that data can take many shapes and levels of complexity. It could be as simple as a shopping list, or as massive and varied as the trillions of interactions happening on social media every day. That’s why we classify data into categories to make it easier to store, manage, and analyse; and two of the most important categories are **structured data** and **unstructured data**.

## **What is structured data?**

When people talk about *structured data*, they're referring to information that's organised in a fixed, predictable way, like files neatly arranged in labelled folders or books placed in a library according to a cataloguing system. In the digital world, this means data is stored in predefined formats with specific fields, making it easy for machines (and people) to search, sort, and analyse.

Think about a spreadsheet: every column has a name (like “First Name,” “Last Name,” “Date of Birth”), and every row contains consistent, matching types of information. This regularity is what makes structured data so powerful because it is predictable and easy to handle.

## **Key characteristics of structured data**

- 1. Predefined schema:** A *schema* is like a blueprint that describes how data is stored and what types of information each field will hold (text, numbers, dates, etc.).
- 2. Consistent formatting:** You know exactly what kind of data goes where. A “Date” column will always have dates in the same format, for example.
- 3. Ease of storage & querying:** Since it follows rules, structured data can be efficiently stored in relational databases (like MySQL, PostgreSQL, Oracle) and retrieved using SQL.
- 4. Highly searchable:** You can run complex queries and get precise results quickly.

## **Common sources of structured data**

- **Transactional databases:** These are data from point-of-sale systems, inventory management, or customer relationship management (CRM) tools.
- **Spreadsheets:** Data gotten from Excel files or Google Sheets.

- **Sensor readings:** Data from IoT devices where measurements are consistently recorded (temperature, humidity, pressure).
- **Financial systems:** Data retrieved from bank records and accounting ledgers.

### **Real-life examples:**

- **E-commerce site:** Each product listing from your favourite online vendor has fields for product name, price, SKU, category, and stock quantity.
- **Hospital records:** Patient ID, date of admission, diagnosis code, and treatment plan are all in specific fields.
- **Airline booking systems:** Passenger name, flight number, departure date, seat assignment.

### **Why structured data matters**

Structured data is the backbone of traditional analytics and reporting. Thanks to its consistency, it's easy to integrate into dashboards, create summaries, and run statistical analyses. Businesses rely heavily on it for day-to-day operations, from sales tracking to performance metrics.

However, and this will set us up for the next section, the world doesn't only produce structured data. A huge portion of the information around us is messy, irregular, and doesn't fit neatly into tables. That's where *unstructured data* comes in.

### **What is unstructured data?**

If structured data is the neat, organised spreadsheet or database table, unstructured data is everything else — the messy, free-flowing, unpolished information that doesn't follow a predefined format. It's the kind of data that

can't be easily stored in rows and columns, and yet it makes up the majority of the information we interact with every day.

Unstructured data is abundant. When you send a text message, post a photo on social media, record a voice note on WhatsApp, or write an email, you're creating unstructured data. When companies collect customer reviews, product images, chat logs, audio recordings, or videos, that's all unstructured data too.

In fact, it's estimated that more than 80% of the world's data is unstructured (you can read more [here](#) and [here](#)), and it's growing exponentially as technology and content creation continue to expand.

### **Key characteristics of unstructured data**

- 1. No fixed format:** Unlike structured data, unstructured data doesn't follow a predefined schema. A single file could be a paragraph of text, a photograph, a video, or even a PDF that contains a mix of text and images.
- 2. Variety and complexity:** Unstructured data comes in many forms: documents, emails, social media posts, audio recordings, videos, web pages, sensor data streams, logs, images, scanned documents, and more. Each type may require a different method of analysis.
- 3. Challenging to search and analyse:** Because it isn't organised in a fixed table, retrieving insights from unstructured data often requires special tools like natural language processing (NLP), image recognition, or custom parsing scripts.
- 4. Rich in context:** Despite being messy, unstructured data can provide insights that structured data cannot. A customer's written feedback or a product photo can reveal sentiment, intent, or context that a numerical

rating alone cannot capture. Simply put, it reveals why something happened, not just what happened.

## **Real-life examples:**

To make this concrete, let's look at some examples you can relate to:

- **Social media posts:** Every tweet, Instagram caption, and LinkedIn update you share contains text, images, emojis, and hashtags. And none of these fits neatly into a database table.
- **Emails and chat logs:** Your inbox is a goldmine of unstructured data: some emails have attachments, some are just text, some include links. Searching across all this data requires text analytics.
- **Multimedia content:** YouTube videos, podcast audio files, photographs, and scanned documents are all unstructured. To extract meaningful data, you often need AI models or specialised software.
- **Logs and sensor feeds:** Server logs, website clickstreams, or IoT sensor data can be technically structured in terms of timestamps and event names, but the content within each event may be unstructured, like free-form text messages, error descriptions, or complex JSON blobs.

## **Why unstructured data matters**

Unstructured data is messy, but it's where a lot of the world's insights are hiding. Here's why it's important:

1. **Rich insights:** While structured data tells you what happened, unstructured data often explains *why*. Customer reviews, social media chatter, and emails can reveal motivations, pain points, and trends that numbers alone can't provide.

2. **Growth and volume:** Businesses today generate massive amounts of unstructured data. Companies that can efficiently process it (think AI-driven recommendations or sentiment analysis) gain a huge competitive edge.
3. **AI and Machine Learning:** Modern AI systems, like recommendation engines, chatbots, and voice assistants, rely heavily on unstructured data. Training a model to understand images, text, or speech requires access to large amounts of unstructured information.
4. **Decision-making:** Organisations increasingly combine structured and unstructured data to make decisions. For example, a retailer may use sales numbers (structured) along with product reviews and customer feedback (unstructured) to plan inventory or marketing campaigns.

## **Challenges of unstructured data**

Working with unstructured data is powerful, but it comes with challenges such as:

- **Storage complexity:** Storing unstructured files often requires object storage systems like Amazon S3, Google Cloud Storage, or Azure Blob Storage, rather than traditional databases.
- **Processing difficulty:** You can't just run a simple SQL query to get answers. You need specialised processing pipelines, search engines, or AI tools to extract meaning.
- **Data quality issues:** Unstructured data can contain errors, duplicates, or irrelevant information, making cleaning and preprocessing essential.
- **Integration challenges:** Combining unstructured data with structured data for analytics or reporting requires thoughtful design and often ETL (Extract, Transform, Load) pipelines.

*So far, we've explored structured and unstructured data. But what about the data that's a little of both? Enter semi-structured data.*

## **What is semi-structured data?**

Semi-structured data is a type of data that doesn't fit neatly into the rigid rows and columns of structured databases, but still has some level of organisation that makes it easier to analyse than purely unstructured data. In other words, it's not completely chaotic because it has labels, tags, or markers that give it meaning, even if the layout isn't rigid.

Think of it as the “middle ground” between structured data (perfectly organised, like an Excel spreadsheet) and unstructured data (completely free-form, like a handwritten diary entry).

Semi-structured data is like your kitchen spice rack. Some jars have neat labels and identical shapes (structured). Others are in random jars with handwritten tags (semi-structured). Still organised enough to use, but not uniform.

### **Key features of semi-structured data**

#### **1. Flexible schema**

Unlike structured data, which demands a rigid format from the start, semi-structured data doesn't require predefined tables. New fields or attributes can be added without breaking anything. This flexibility is essential in modern applications where data evolves rapidly.

**Example:** Imagine an e-commerce site storing product data in JSON.

Initially, each product has:

```
{  
  "product_id": 101,  
  "name": "Wireless Headphones",  
  "price": 79.99  
}
```

Later, you want to track **discounts** and **ratings**. With a flexible schema, you can add these fields without affecting the existing data:

```
{  
  "product_id": 101,  
  "name": "Wireless Headphones",  
  "price": 79.99,  
  "discount": 10,  
  "ratings": [5, 4, 5, 3]  
}
```

You can see here that the dataset immediately adapts because the table isn't rigid.

## 2. Self-describing

Semi-structured data often contains metadata or labels that describe the content. This makes it easier to interpret than raw, unstructured data. For example, a JSON object might include field names like “name,” “email,” or “timestamp” that tell you exactly what each piece of data represents.

**Example:** A JSON object representing a user:

```
{  
  "name": "Jane Doe",  
  "email": "jane@example.com",  
  "signup_date": "2025-08-15"  
}
```

Each **key** (`name`, `email`, `signup_date`) describes the data it holds, so even without external documentation, you can understand the meaning immediately.

Now, compare this with raw unstructured text below:

```
Jane Doe, jane@example.com, 08/15/2025
```

Without labels, you'd have to guess which piece is the name, email, or date.

### 3. Partial organisation

Semi-structured data has a hierarchy or nested relationships. It's not completely free-form because elements exist between elements and are organised in a way that machines and humans can leverage for querying and analysis.

#### Example:

A JSON representation of an order:

```
{  
  "order_id": 5001,
```

```
"customer": {  
    "name": "John Smith",  
    "email": "john@example.com"  
},  
"items": [  
    {"product": "Laptop", "quantity": 1, "price": 1200},  
    {"product": "Mouse", "quantity": 2, "price": 25}  
]  
}
```

Here, `customer` and `items` create **nested relationships**. You can easily extract all items, calculate totals, or find the customer email; something much harder when working with completely unstructured data.

## 4. Human-readable

Many semi-structured formats, like XML, JSON, or YAML, are text-based, which means you can read and understand them without special tools.

### Example:

A simple XML snippet for a blog post:

```
<blogPost>  
    <title>Introduction to Data Engineering</title>  
    <author>Wuraola Ademola-Shanu</author>  
    <publishedDate>2025-08-15</publishedDate>  
    <tags>  
        <tag>Data Engineering</tag>  
        <tag>Beginners</tag>  
    </tags>  
</blogPost>
```

Even without software, you can clearly see the title, author, date, and tags. You can read, modify, or debug this manually, unlike a binary file or raw log

data.

## Common formats of semi-structured data

Format	Description	Example/Use case
JSON (JavaScript Object Notation)	Lightweight, text-based format using key-value pairs.	Web APIs, configuration files, and NoSQL databases.
XML (eXtensible Markup Language)	Tag-based hierarchical format.	Data exchange between systems, document storage.
YAML(YAML Ain't Markup Language)	Human-readable format for structured configurations.	DevOps pipelines, application settings.
Avro / Parquet / ORC	Binary formats optimised for big data.	
Emails	Text-based content with structured headers (to, from, date) and an unstructured body.	Communication archives, automated processing.

Table comparing common data formats. Columns are 'Format', 'Description', and 'Example/Use case'. Formats listed are: JSON (JavaScript Object Notation), XML (eXtensible Markup Language), YAML (YAML Ain't Markup Language), Avro/Parquet/ORC, and Emails. Descriptions and example use cases include web APIs, configuration files, data exchange between systems, DevOps pipelines, big data, and communication archives.

## Examples of semi-structured data

### 1. E-commerce product information

Online stores often store product details in JSON. Each product might include a name, price, categories, stock levels, and optional fields like discount codes or reviews. The structure is present, but it can vary across products.

```
{  
  "product_id": 101,  
  "name": "Bluetooth Headphones",  
  "price": 79.99,  
  "categories": ["Electronics", "Audio"],  
  "discount": 10,  
  "stock": {"warehouse": 50, "store": 20}  
}
```

## 2. Social media APIs

Platforms like Twitter, Facebook, or LinkedIn return post data in JSON. Each post has fields like “user,” “timestamp,” “text,” and optional “media” or “comments,” making it partially organised but not rigidly structured.

## 3. IoT devices

Smart home sensors (thermostats, motion detectors) or fitness trackers send semi-structured JSON or XML data to cloud platforms. Sure, the data has labels (temperature, heart rate, timestamp), but it isn’t confined to a table.

Get Wuraola Ademola-Shanu’s stories in your inbox

Join Medium for free to get updates from this writer.

Enter your email

Subscribe

## 4. Emails and logs

Emails contain structured headers (sender, recipient, subject) and semi-structured content (body text, attachments). Server logs may have timestamps and event codes with free-form messages.

## Structured vs. unstructured vs. semi-structured data(comparison table)

Feature	Structured data	Semi-structured data	Unstructured data
Shape	Neatly organized in fixed fields and rows, following a rigid schema.	Organized but flexible. Has tags, markers, or metadata, but no strict schema.	No predefined structure; content is free-form.
Examples	Relational databases, CRM systems, inventory spreadsheets.	JSON, XML, CSV with irregular fields, sensor logs, NoSQL documents.	Images, videos, emails, social media posts, audio recordings.
Storage	Relational databases (SQL), data warehouses.	NoSQL databases, object stores, data lakes.	File systems, content management systems, object stores, data lakes.
Typical tools	SQL-based tools (MySQL, PostgreSQL, Oracle), BI dashboards (Tableau, Power BI).	NoSQL DBs (MongoDB, Couchbase), big data tools (Hadoop, Spark), ETL pipelines.	AI/ML tools, NLP frameworks, multimedia processing software.
Best for	Transactions, financial records, structured reporting, regulatory compliance.	Flexible data exchange, IoT, APIs, evolving data models.	Rich media, customer feedback, creative assets, pattern recognition.
Limitations	Inflexible, as schema changes can be costly.	Can get messy if metadata is inconsistent.	Hard to search/analyze without preprocessing. It's storage-heavy.

Table comparing structured, semi-structured, and unstructured data across attributes such as shape, examples, storage, typical tools, use cases, and limitations.

## Common pitfalls beginners should watch out for

Even the best data engineers started somewhere, and it's easy to stumble when you're just getting familiar with structured, unstructured, and semi-structured data. Here are some common traps beginners fall into and how to avoid them:

**Treating everything like a table:** It's tempting to shove all your data into neat rows and columns because spreadsheets feel safe. But messy texts, images, or logs often don't fit into a rigid table. For example, trying to store customer reviews full of emojis, slang, and line breaks in a spreadsheet can make it unreadable and hard to analyse. Think of it like trying to fit a jigsaw puzzle piece into the wrong spot. It just won't work! 

**Ignoring metadata or labels for unstructured data:** Metadata is like sticky notes telling you what's inside a file. Skipping them might not seem bad at first, but later, finding a single image, email, or log in a sea of files becomes a nightmare. Imagine snapping hundreds of photos on your phone without adding any albums or tags. Hours of scrolling later, you still can't find "that one picture from last week." A little organisation upfront saves hours of frustration later.

**Skipping data quality checks for semi-structured logs:** Semi-structured data like JSON or XML can look tidy at first, but keys can drift, optional fields can pop up, or timestamps can get messy. For instance, a fitness tracker might log heart rate in beats per minute on some days, and seconds on others. If you don't check, your weekly average becomes meaningless. Without regular checks, you could be analysing garbage without even realising it.

**Over-engineering storage before asking the right questions:** It's easy to get excited and start designing a massive, complicated storage system. But before you do, pause and ask: *What questions do I actually need to answer with this data?* For example, creating a huge multi-layered database for your personal notes app when all you need is a simple search function is overkill. Building storage without a clear purpose is like constructing a mansion when all you really need is a cosy apartment.

**Pro tip:** Start simple, stay organised, and always think about the story your data needs to tell. That's how beginners avoid getting lost in the chaos (at least, that's what worked for me).

## **FAQs:**

### **1. Is a PDF structured or unstructured?**

Mostly unstructured, because PDFs are often just “snapshots” of text and images. You can’t easily query them like a database. Sometimes, they include tables or tagged sections that create semi-structured chunks—like a messy notebook with a few neat lists tucked inside.

### **2. Is a CSV structured?**

Usually yes. Rows and columns make it structured...until they don't.

Columns can drift, headers can be missing, or data types can mix. It's like a spreadsheet with someone else's messy edits. You still have a table, but it's not perfect.

### **3. Where does JSON fit?**

Semi-structured. JSON has labels (keys) describing the data, but the structure can vary across entries. For example, one product might have a “discount” field and another might not.

### **4. File types & formats:**

- Is an Excel file structured or semi-structured? (usually structured, but formulas or merged cells can complicate things).
- Are images or videos ever structured? (mostly unstructured).
- What about logs? CSV logs? JSON logs? (semi-structured).

## 5. Do I need different databases?

Often, yes. Different types of data thrive in different storage systems:

- **Structured data:** Relational databases (like MySQL, PostgreSQL) are perfect for rows and columns.
- **Semi-structured data:** NoSQL databases (like MongoDB) or JSON stores work best.
- **Unstructured data:** Object storage (like Amazon S3, Google Cloud Storage) handles images, videos, and PDFs.

## 6. Can one system handle everything?

Technically, yes, but it can get messy. Mixing structured and unstructured data in the same database is like trying to store furniture, clothes, and food in one (tiny) closet. It's possible, but messy and slow

## 7. How do I decide which database to use?

Ask yourself: *What questions do I need to answer? How will I query this data? Do I need speed or flexibility?* Your answers guide whether you pick a warehouse, a NoSQL store, or an object storage solution.

## 8. Practical considerations:

- Which type of data grows fastest? (unstructured, especially images, videos, social media content).
- Which type is easiest to analyse as a beginner? (structured data).
- Which is most likely to contain hidden insights? (unstructured data, if you know how to process it).

## **9. Can unstructured data become semi-structured or structured?**

Absolutely. With the right tools, like OCR for text in images, NLP for emails or chat logs, or tagging and parsing pipelines, you can extract structured insights from unstructured data. Think of it as tidying your messy closet into labelled bins. You still have all your stuff, but now it's usable.

## **10. Why do semi-structured formats exist at all?**

Because the world isn't perfectly neat. Semi-structured formats like JSON, XML, or YAML give machines enough organisation to process data, while keeping flexibility for real-world, messy data.

## **11. Any beginner tips for handling all three types?**

- **Start simple:** Don't overcomplicate storage.
- **Tag everything:** Metadata is your friend.
- **Validate regularly:** Semi-structured logs and tables can drift.
- **Choose tools wisely:** Match your database to your data type.

**Remember the story:** Data only has value when it answers real questions.

Absolutely! Based on the full article you provided, here's an exhaustive glossary that's beginner-friendly, comprehensive, and human, covering all key terms, file types, storage options, and concepts mentioned:

## **Glossary: Structured, Semi-Structured & Unstructured Data**

Data

- **Definition:** Information collected and stored in a way that can be referenced, analysed, and used to make decisions.
- **Examples:** Numbers, words, images, audio, clicks, sensor readings, or measurements.
- **Tip:** Data is everywhere. Even checking your phone or taking a photo creates it.

## Structured data

- **Definition:** Data organised in a fixed, predictable format (rows, columns, tables).
- **Key features:** Predefined schema, consistent formatting, easy to search and query.
- **Common sources:** Spreadsheets, relational databases, transactional systems, IoT sensors.
- **Examples:** Excel files, CRM tables, airline bookings, hospital records.
- **Best for:** Traditional analytics, dashboards, reporting, business operations.

## Semi-structured data

- **Definition:** Data that has some organisation (tags, labels, or keys) but doesn't fit rigid rows and columns.
- **Key features:** Flexible schema, self-describing, partial organisation, human-readable.
- **Common formats:** JSON, XML, YAML, Avro, Parquet, ORC, emails, API outputs.

- **Examples:** Product info in JSON, social media API data, IoT device logs, emails.
- **Best for:** Applications that require flexibility but still need some structure for processing.

## Unstructured data

- **Definition:** Messy, free-form data with no predefined format or schema.
- **Key Features:** No fixed format, highly varied, hard to search/analyse, rich in context.
- **Common formats:** Text documents, images, videos, audio files, PDFs, chat logs, web pages.
- **Examples:** Social media posts, YouTube videos, scanned documents, server logs.
- **Best for:** AI/ML, sentiment analysis, deep insights, multimedia content analysis.

## Metadata

- **Definition:** Descriptive information about data that makes it easier to find, manage, and interpret.
- **Example:** File tags, headers in emails, labels in JSON objects.
- **Tip:** Skipping metadata is a common beginner mistake — it makes searching chaotic.

## Schema

- **Definition:** A blueprint that defines the structure and type of data in a dataset.
- **Example:** Column names and types in a spreadsheet or database table.
- **Tip:** Structured data relies on rigid schemas; semi-structured data is flexible.

## ETL (Extract, Transform, Load)

- **Definition:** Process of moving and transforming data from source to destination for analysis.
- **Tip:** Useful for combining structured and unstructured data for reporting or dashboards.

## Relational Database

- **Definition:** A database storing structured data in tables with predefined columns.
- **Examples:** MySQL, PostgreSQL, Oracle.
- **Best for:** Predictable, highly organised data with complex queries.

## NoSQL Database

- **Definition:** A database designed for semi-structured or unstructured data, often schema-flexible.
- **Examples:** MongoDB, Couchbase.
- **Best for:** JSON, XML, big data pipelines, rapidly evolving datasets.

## Object Storage

- **Definition:** A storage system optimised for unstructured data like images, videos, and PDFs.
- **Examples:** Amazon S3, Google Cloud Storage, Azure Blob Storage.
- **Tip:** Not ideal for SQL-style queries, but perfect for large media and documents.

## CSV (Comma-Separated Values)

- **Definition:** Tabular structured data stored as plain text, where commas separate fields.
- **Tip:** Usually structured, but can become messy if columns drift or headers are missing.

## JSON (JavaScript Object Notation)

- **Definition:** Text-based format for semi-structured data using key-value pairs.

### Example:

```
{  
  "product_id": 101,  
  "name": "Wireless Headphones",  
  "price": 79.99  
}
```

## XML (eXtensible Markup Language)

- **Definition:** Semi-structured, text-based format using nested tags to organise data.

### Example:

```
<user>
  <name>Landon Carter</name>
  <email>landonctr@yahoo.com</email>
</user>
```

## YAML (YAML Ain't Markup Language)

- **Definition:** Human-readable semi-structured format often used for configuration files.

## Log Data

- **Definition:** Records of events generated by systems, apps, or devices.
- **Tip:** Can be structured (timestamps) but often semi-structured or unstructured (free text).

## PDF (Portable Document Format)

- **Definition:** Mostly unstructured; sometimes contains semi-structured tables or tagged sections.

## Data Quality Checks

- **Definition:** Ensuring data is accurate, consistent, and usable before analysis.

## Conclusion

By now, you can probably spot structured, semi-structured, and unstructured data in your everyday life...from the spreadsheet you use at work, to the photos on your phone, and the emails in your inbox. Data isn't some mysterious tech-only concept anymore; it's all around you, and understanding how it's organised is the first step to making it useful.

And hey, if there's a concept that still feels confusing and you want me to explain better, drop a comment. I'd love to help break it down in a way that actually clicks for you.

*Like what you just read? Don't just nod and move on. Subscribe! That way, you will never have to frantically Google "where did I read that great article?" again.*

😊 No spam, no fluff, just solid, useful content delivered straight to your inbox.

Data Engineering

Structured Data

Unstructured Data

Data

Tech For Beginners



Written by Wuraola Ademola-Shanu

193 followers · 8 following

Follow