# 

# Introduction to Text Analysis in Elasticsearch

## 1. Overview of Text Analysis

Text analysis is the process Elasticsearch uses to transform and store text values in an efficient data structure for searching. It ensures that text data can be searched and queried effectively by processing it during indexing.

When a text value is indexed, it undergoes analysis, which involves breaking it down and converting it into a format that is optimized for searching. The exact values you see in the '\_source' key are not the same as those used internally for search queries.

## 2. The Role of Analyzers

Analyzers play a central role in processing text during indexing. They are composed of three components:  
 - \*\*Character Filters\*\*  
 - \*\*Tokenizer\*\*  
 - \*\*Token Filters\*\*  
The output of an analyzer is a searchable data structure.

## 3. Components of an Analyzer

### 3.1 Character Filters

Character filters process the raw text by adding, removing, or modifying characters.

An analyzer can have zero or more character filters, which are applied in the specified order.

Example: The 'html\_strip' filter removes HTML elements and converts HTML entities.

### 3.2 Tokenizer

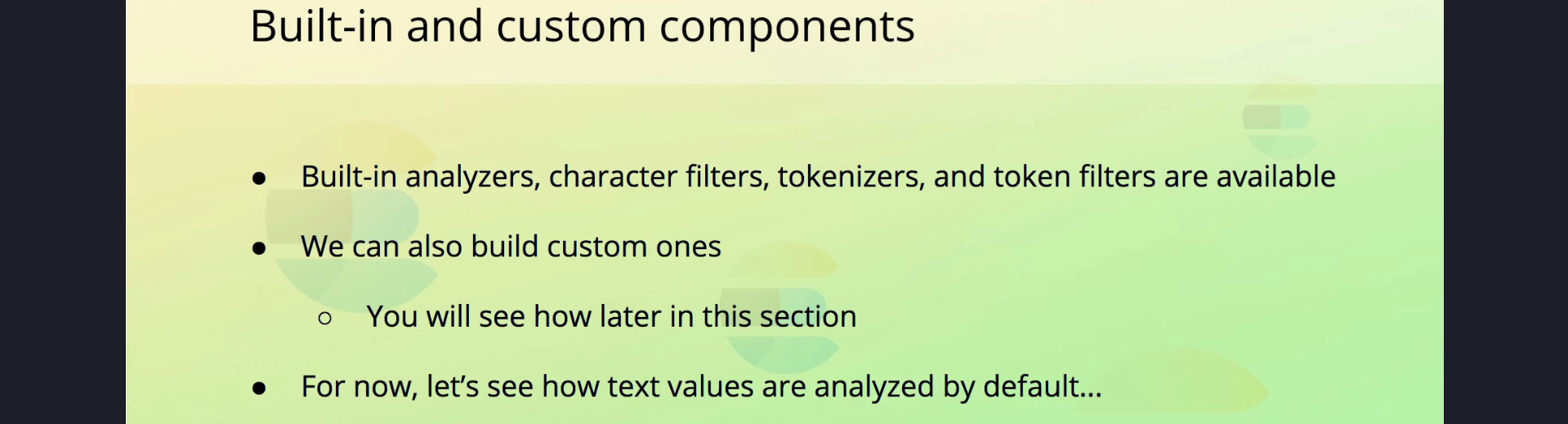
A tokenizer splits the text into smaller units called tokens.   
Each analyzer must have exactly one tokenizer.

**Example**: A tokenizer can split a sentence into words by splitting on whitespace, punctuation, or other delimiters.

**Note**: The tokenizer also records the character offsets of each token in the original text. This will be discussed in detail later.

### 3.3 Token Filters

Token filters modify the tokens produced by the tokenizer.   
Like character filters, an analyzer can have zero or more token filters applied in sequence.

**Example**: The 'lowercase' token filter converts all tokens to lowercase.  
  


## 4. Default Behavior and Standard Analyzer

By default, Elasticsearch uses the 'standard' analyzer for all 'text' fields unless configured otherwise.

The 'standard' analyzer behaves as follows:  
 - No character filter is applied by default.  
 - The tokenizer splits text into tokens using the **Unicode Segmentation algorithm**, which breaks sentences into words based on whitespace, hyphens, and similar delimiters.  
 - Punctuation such as commas, periods, and exclamation marks are removed during tokenization.  
 - The 'lowercase' token filter converts all tokens to lowercase.

## 5. Custom Analyzers

Elasticsearch allows you to create custom analyzers by combining character filters, tokenizers, and token filters. This flexibility enables you to tailor the analysis process to specific use cases.

While the 'standard' analyzer is sufficient for most scenarios, understanding how to build custom analyzers can be invaluable for specialized requirements.

## 6. Summary

Text analysis is a foundational concept in Elasticsearch that transforms text into a searchable format. It involves the use of analyzers, which consist of character filters, a tokenizer, and token filters.

The 'standard' analyzer is the default for text fields and is typically sufficient for most use cases. However, custom analyzers can be created by combining built-in components to suit specific needs.

In this section, we introduced the basic concepts of text analysis and the components of analyzers. Further details on analyzers and their customization will be covered later.