# Regular Expression Operator ($regex)

Regular expressions allow you to match strings in documents based on pattern matching. They are very useful for flexible string searches, like finding substrings or case-insensitive matches.

## Regular Expressions Overview

| Operator | Description | Example Usage |
|---|---|---|
| **$regex** | Matches strings using a regular expression pattern | { name: { $regex: /^Smart/, $options: 'i' } } |

## $regex

- Matches documents where the specified field matches a **string pattern**.
- Patterns can be specified as a RegExp object (/pattern/) or a string.
- $options allows modifying regex behavior (like case-insensitive matching).
- Common options:
  - i = case-insensitive match
  - m = multiline mode
  - x = ignore whitespace in pattern

**Example 1: Find products whose name starts with "Smart" (case-insensitive)**

```
db.products.find({
 name: { $regex: /^Smart/, $options: 'i' }
});
```

- *Finds products with names starting with "Smart", ignoring case.*

**Example 2: Find products with "key" anywhere in the name (case-insensitive)**

```
db.products.find({
 name: { $regex: "key", $options: 'i' }
});
```

- *Finds products that contain the substring "key" in their name, ignoring case.*