



# MangoDB Class

**#MangoDB Notes**

# String Aggregation Operators



# String Aggregation Operators

- \$concat
- Concatenates strings and returns the concatenated string.
- Syntax:  
{ \$concat: [ <expression1>, <expression2>, ... ] }



# Product Collection Data

```
mydb> db.product.find()
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    desc: 'output device'
  }
]
mydb>
```



# \$concat Example

```
mydb> db.product.aggregate([{$project:{productDetail:{$concat:["$name","-","$desc"]}}}] )
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    productDetail: 'keyboard-input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    productDetail: 'mouse-input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    productDetail: 'printer-output device'
  }
]
mydb>
```



# \$indexOfBytes

- Searches a string for an occurrence of a substring and returns the UTF-8 byte index (zero-based) of the first occurrence. If the substring is not found, returns -1.
- **Syntax:**  
{ \$indexOfBytes: [ <string expression>, <substring expression>, <start>, <end> ] }



# Product Collection Data

```
mydb> db.product.find()
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    desc: 'output device'
  }
]
mydb>
```



# \$indexOfBytes Example

- The following operation uses the \$indexOfBytes operator to retrieve the indexes at which the string dev is located in each item:

```
mydb> db.product.aggregate([{$project:{byteLocation:{$indexOfBytes:["$desc","dev"]}}}]  
[  
  { _id: ObjectId("618e574632d48ca345e03565"), byteLocation: 6 },  
  { _id: ObjectId("618e574632d48ca345e03566"), byteLocation: 6 },  
  { _id: ObjectId("618e574632d48ca345e03567"), byteLocation: 7 }  
]  
mydb>
```





# \$split

- Divides a string into an array of substrings based on a delimiter. \$split removes the delimiter and returns the resulting substrings as elements of an array.
- If the delimiter is not found in the string, \$split returns the original string as the only element of an array.
- **Syntax:**  
{ \$split: [ <string expression>, <delimiter> ] }



# Product Collection Data

```
mydb> db.product.find()
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    desc: 'output device'
  }
]
mydb>
```



# \$split Example

```
mydb> db.product.aggregate([{$project:{Details:{$split:["$desc"," "]]}}])
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    Details: [ 'input', 'device' ]
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    Details: [ 'input', 'device' ]
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    Details: [ 'output', 'device' ]
  }
]
mydb>
```



# \$strcasecmp

- Performs case-insensitive comparison of two strings. Returns
  - 1 if first string is “greater than” the second string.
  - 0 if the two strings are equal.
  - -1 if the first string is “less than” the second string.
- **Syntax:**  
**{ \$strcasecmp: [ <expression1>, <expression2> ] }**



# productDetails Collection Data

```
mydb> db.productDetails.find()
[
  { _id: 1, name: 'ABC', quarter: '13Q1' },
  { _id: 2, name: 'XYZ', quarter: '13Q4' },
  { _id: 3, name: 'PQR', quarter: '13Q2' }
]
mydb>
```



# \$strcasecmp Example

```
mydb> db.productDetails.aggregate([{$project:{name:1,result:{$strcasecmp:["$quarter","13Q4"]}}}]])
[
  { _id: 1, name: 'ABC', result: -1 },
  { _id: 2, name: 'XYZ', result: 0 },
  { _id: 3, name: 'PQR', result: -1 }
]
mydb>
```



# \$strLenBytes

- Returns the number of UTF-8 encoded bytes in the specified string.
- **Syntax:**  
`{ $strLenBytes: <string expression> }`



# Product Collection Data

```
mydb> db.product.find()
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    desc: 'output device'
  }
]
mydb>
```





# \$strLenBytes Example

```
mydb> db.product.aggregate([{$project:{name:1,length:{$strLenBytes:"$name"}}}])
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    length: 8
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    length: 5
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    length: 7
  }
]
mydb>
```



# \$substrBytes

- Returns the substring of a string. The substring starts with the character at the specified UTF-8 byte index (zero-based) in the string and continues for the number of bytes specified.
- **Syntax:**  
**{ \$substrBytes: [ <string expression>, <byte index>, <byte count> ] }**



# Product Collection Data

```
mydb> db.product.find()
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    desc: 'output device'
  }
]
mydb>
```



# \$substrBytes Example

```
mydb> db.product.aggregate([{$project:{name:1,substrDetail:{$substrBytes:["$name",0,3]}}}])
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    substrDetail: 'key'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    substrDetail: 'mou'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    substrDetail: 'pri'
  }
]
mydb>
```



# \$toLower

- Converts a string to lowercase, returning the result.
- **Syntax:**  
`{ $toLower: <expression> }`



# productDetails Collection Data

```
mydb> db.productDetails.find()
[
  { _id: 1, name: 'ABC', quarter: '13Q1' },
  { _id: 2, name: 'XYZ', quarter: '13Q4' },
  { _id: 3, name: 'PQR', quarter: '13Q2' }
]
mydb>
```



# \$toLower Example

```
mydb> db.productDetails.aggregate([{$project:{name:1,lwrName:{$toLower:"$name"}}}])
[
  { _id: 1, name: 'ABC', lwrName: 'abc' },
  { _id: 2, name: 'XYZ', lwrName: 'xyz' },
  { _id: 3, name: 'PQR', lwrName: 'pqr' }
]
mydb>
```



# \$toUpper

- Converts a string to uppercase, returning the result.
- **Syntax:**  
`{ $toUpper: <expression> }`





# Product Collection Data

```
mydb> db.product.find()
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    desc: 'input device'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    desc: 'output device'
  }
]
mydb>
```



# \$toUpper Example

```
mydb> db.product.aggregate([{$project:{name:1,uprName:{$toUpper:"$name"}}}])
[
  {
    _id: ObjectId("618e574632d48ca345e03565"),
    name: 'keyboard',
    uprName: 'KEYBOARD'
  },
  {
    _id: ObjectId("618e574632d48ca345e03566"),
    name: 'mouse',
    uprName: 'MOUSE'
  },
  {
    _id: ObjectId("618e574632d48ca345e03567"),
    name: 'printer',
    uprName: 'PRINTER'
  }
]
mydb>
```

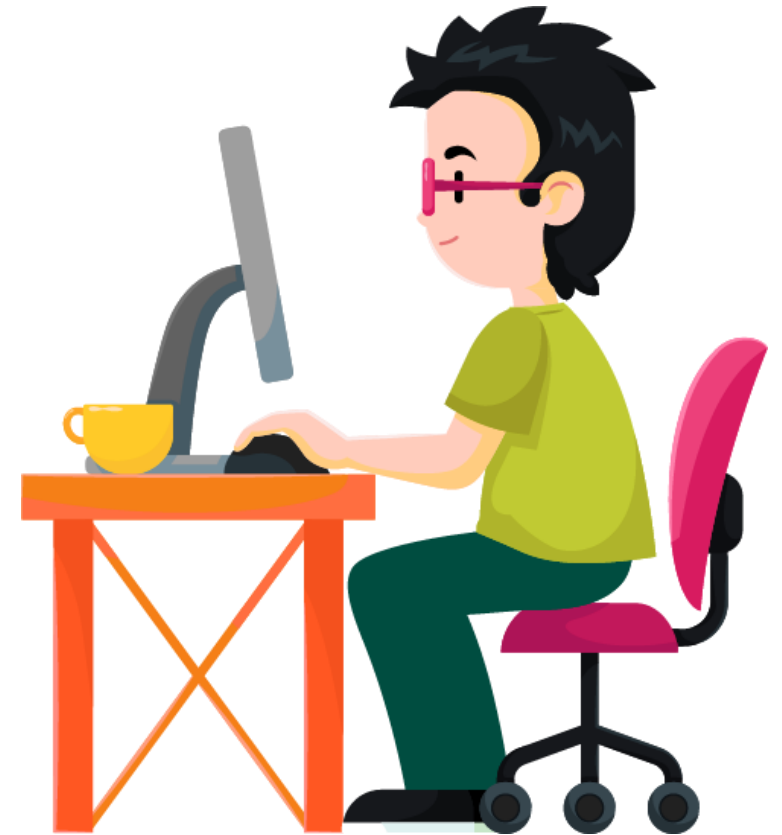


# Get Exclusive Video Tutorials



[www.apptutorials.com](http://www.apptutorials.com)

<https://www.youtube.com/user/Akashtips>





Get More Details

[www.akashsir.com](http://www.akashsir.com)



# If You Liked It !

## Rating Us Now



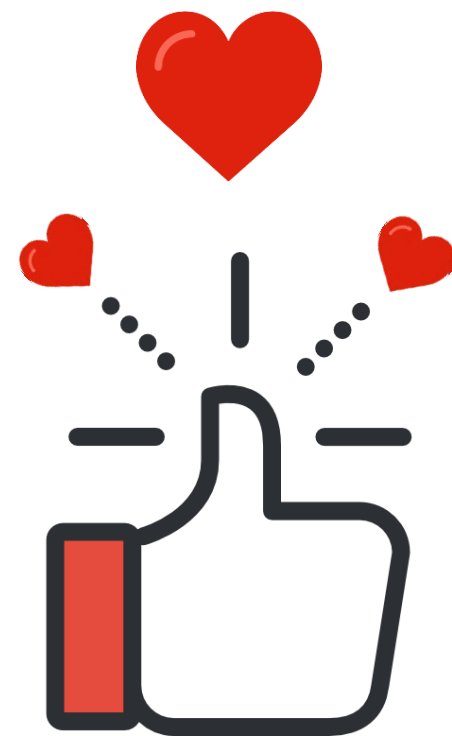
**Just Dial**

[https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4\\_BZDET](https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET)



**Sulekha**

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



# Connect With Me



Akash Padhiyar  
#AkashSir

[www.akashsir.com](http://www.akashsir.com)

[www.akashtechlabs.com](http://www.akashtechlabs.com)

[www.akashpadhiyar.com](http://www.akashpadhiyar.com)

[www.aprtutorials.com](http://www.aprtutorials.com)

## # Social Info



Akash.padhiyar



Akashpadhiyar



Akash\_padhiyar



+91 99786-21654



#Akashpadhiyar

#aprtutorials