

# MongoDB

**עדכון מסמכים**

**מחיקת מסמכים**

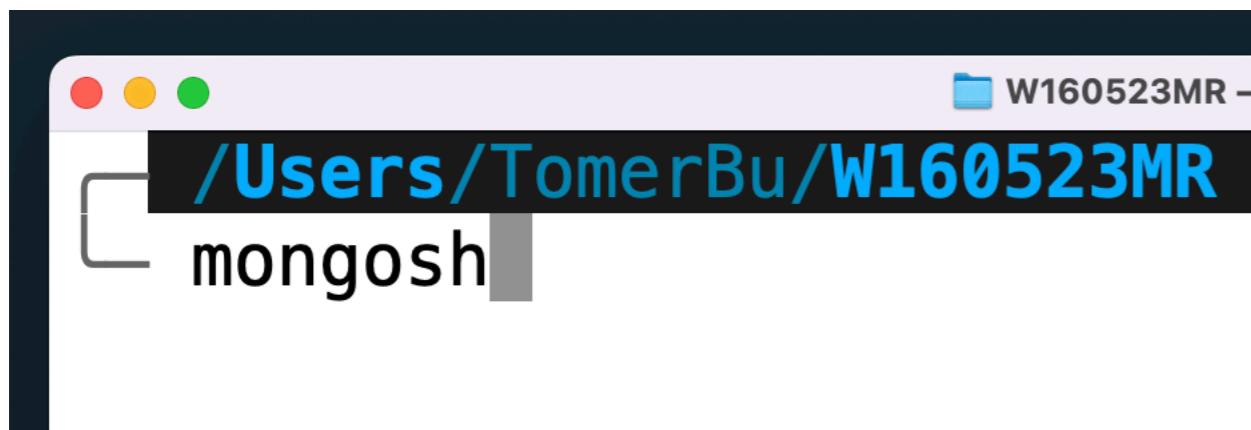
**חיפוש מסמכים**

**ביטויים רגולריים - Regex**

TomerBu

# Mongosh

אפשר לעבוד עם Mongosh במקומם



```
| test> use lec2
```

```
| lec2> db.myCollection.insertOne({})
```

# דטה ביחס לדוגמאות הבאות:

```
db.users.insertMany([
```

<https://jsonplaceholder.typicode.com/users>

```
> use lec2
< switched to db lec2
lec2> db.users.insertMany([
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
```

# עדכו רשומות במשק גרافي:

```
3  name: "Leanne Grahams"          String
4  username: "Bret"                String
5  email: "Sincere@april.biz"     String
6  ▶ address: Object              Object
7  phone: "1-770-736-8031 x56442" String
8  website: "hildegard.org"       String
9  ▶ company: Object              Object
```

Document modified.

CANCEL

UPDATE



# עדכון Documents עם המתודה updateOne המתודה מקבלת 2 פרמטרים:

db  
collection  
documents

```
db.myCollection.updateOne(  
  <filter>, ←  
  <update> ←  
)
```

אובייקט: איזה מסמך רוצים לעדכן:

אובייקט: המידע המעודכן - דוש אופרטור:

```
db.users.updateOne({id: 2}, {$set: {name: "Elvin Howell"}})
```

update users set name='elvin' WHERE id = 2

# מיני תרגיל:

```
▼  {
  ▶    "_id": { ... },
  ▶    "id": 1,
  ▶    "name": "Leanne Graham",
  ▶    "username": "Bret",
  ▶    "email": "Sincere@april.biz",
  ▶    "address": { ... },
  ▶    "phone": "1-770-736-8031 x56442",
  ▶    "website": "hildegard.org",
  ▶    "company": { ... }
}
```

עבור משתמש עם id של 1

שנו את השם לשם אחר לבחירתכם

שנו את שם המשתמש

שנו את כתובת האימייל

כרגיל - אושמה לפתרון שלכם בצ'אט - יעזר למי שרצה להשוות  
ואצבע אם סיימתם

```
db.users.updateOne({id: 2}, {$set: {name: "Elvin Howell"}})
```

# פתרונות לתרגילים:

```
db.users.updateOne({id: 2}, {$set: {name: "Elvin Howell"}})
```

```
db.users.updateOne({id: 1}, {$set: {name: "moe", username: "moemoe", email: "moe@gmail.com"}})
```

ctrl+r  
מראנו

# קישורים לדוקומנטציה:

update documents Mongo

## הדרכה - Tutorial

<https://www.mongodb.com/docs/manual/tutorial/update-documents/>

## כל המידע על פונקציה - Reference

<https://www.mongodb.com/docs/manual/reference/method/db.collection.updateOne/#mongodb-method-db.collection.updateOne>

מיועד למי שמכיר את המתודה ורוצה לחזור דקויות

# וסף לדוגמאות הבאות:

```
lec2> db.inventory.insertMany( [  
    { item: "canvas", qty: 100, size: { h: 28, w: 35.5,  
    { item: "journal", qty: 25, size: { h: 14, w: 21, uo  
    { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uo  
    { item: "mousepad", qty: 25, size: { h: 19, w: 22.85  
    { item: "notebook", qty: 50, size: { h: 8.5, w: 11,  
    { item: "paper", qty: 100, size: { h: 8.5, w: 11, uo  
    { item: "planner", qty: 75, size: { h: 22.85, w: 30,  
    { item: "postcard", qty: 45, size: { h: 10, w: 15.25  
    { item: "sketchbook", qty: 80, size: { h: 14, w: 21,  
    { item: "sketch pad", qty: 95, size: { h: 22.85, w:  
] );|
```

# עדכו שדה באובייקט מוכן:

## Update fields in nested Objects:

```
db.inventory.updateOne({item: "journal"}, {$set: {"size.uom": "in"}})
```

↑  
פילטר

↑      ↑  
שדה באובייקט    אובייקט

# עדכון לפי `_id`:

לרוב נרצה לעדכן רשומות לפי מזהה ייחודי  
(לא נרצה לעדכן לפי שם או לפי כמות - כי זה עלול לעדכן אלמנט אחר מזה שתכננו)

```
{
  "_id": {
    "$oid": "6562fae383d29d14e6eb2ed4"
  },
  "item": "canvas",
}
```

```
db.inventory.updateOne({_id: ObjectId('6562fae383d29d14e6eb2ed4')}, {$set: {item: 'canvas2'}})
```

↑              ↑  
      פילטר:      מופיע של (...):

# מיני תרגיל עדכו לפי `_id`:

מINI תרגיל:

שנו לפি `_id` את המסמך של paper

שנו את qty ל 99

שנו את הגובה ל 10

שנו את הסטטוס ל "E"

```
_id: ObjectId('63c8f7efe9f4f0db45a0eb77')
item: "paper"
qty: 100
▼ size: Object
  h: 8.5
  w: 11
  uom: "cm"
status: "D"
```

# מיני תרגיל עדכו לפי `:_id`

```
db.inventory.updateOne({_id: ObjectId('6562fae383d29d14e6eb2ed9')},  
{  
  $set:{  
    qty: 99,  
    "size.h": 10,  
    status: "E"  
  }  
})
```

```
_id: ObjectId('63c8f7efe9f4f0db45a0eb77')  
item: "paper"  
qty: 100  
▼ size: Object  
  h: 8.5  
  w: 11  
  uom: "cm"  
status: "D"
```

```
db.inventory.updateOne({  
  _id: ObjectId('6562fae383d29d14e6eb2ed9') },  
{  
  $set:{  
    qty: 99,  
    "size.h": 10,  
    status: "E"  
  }  
})  
[]
```

# הפעולה replaceOne

## מחליפה את כל המסמך (לא כולל `_id`)

```
db.ads.insertOne({title: 'the title', description: 'some description'})
```

```
lec2> db.ads.replaceOne({  
    _id: ObjectId('6563094083d29d14e6eb2ede')  
}, {  
    comment: "some comment"  
})
```

```
▼ {  
▼ "_id": {  
    "$oid": "6563094083d29d14e6eb2ede"  
},  
"comment": "some comment"  
}
```

# הפעולה updateOne

מוסיפה או מעדכנת אובייקטים  
מוסיפה או מעדכנת ערך של שדות

```
db.users.updateOne({id: 11}, {$set: {name: 'dave'}}, {upsert: true})
```



**מוסיפה אובייקט חדש**

```
db.users.updateOne({id: 1}, {$set: {somekey: 'key'}})
```



**מוסיפה או מעדכנת ערך של שדות**

# מבוא לשינוי ערך במערך:

```
db.inventory.replaceOne()  
{ item: "paper" },  
{ item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 40 } ] }  
()
```

# שינויי של ערך במערך לפי אינדקס:

```
db.inventory.updateOne({item: 'paper'}, {  
    $set: {  
        "instock.0" : {warehouse: "C", qty: 100}  
    }  
})
```

אינדקס שם המערך

שינויי של האיבר הראשון במערך (לפי אינדקס)

# מיני תרגיל:

(1) רוקנו את האוסף של `Inventory`

(2) הוסיפו לאוסף את הרשומות הבאות:

```
db.inventory.insertMany([
  { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
  { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
  { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
  { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
  { item: "postcard", qty: 45, tags: ["blue"] }, dim_cm: [ 10, 15.25 ] }
]);
```

(3) שנו את `tags` של `journal` שמדובר הראשון במערך יהיה "color"

# פתרונות:

```
lec2> db.inventory.updateOne({_id: ObjectId('6563119483d29d14e6eb2edf')},  
{ $set: []  
  'tags.0': 'color'  
}  
}  
)
```

## עדכון של הרבה מסמכים יחד: **updateMany**

```
פיטר ריק (כל המסמכים)           ↓           שדה שהוספנו           ↓  
|db.users.updateMany({}, { $set: {lastUpdate: new Date()}})
```

נלמד עוד פיטרים בהמשך השיעור היום

# מחיקת מסמכים:

הפרמטר הוא filter (כמו update)

deleteOne  
↓

```
db.inventory.deleteOne({ 'name': 'paper' })
```

deleteMany  
↓

```
db.inventory.deleteMany({ 'item': 'paper' })
```

deleteOne by id:  
↓

```
db.inventory.deleteOne({_id: ObjectId('6563119483d29d14e6eb2edf'))})
```

필טר ריק - כל האלמנטים -  
↓

```
lec2 > db.inventory.deleteMany()
```

מראקן את האוסף

# תרגילים:

**אשמה לפתרונות בצ'אט.  
מי שימושים תרגיל שיחלה.**

## משימה 1

- צור אוסף חדש בשם users עבור בסיס נתונים חדש בשם abtest2
- הוסף לאוסף מספר מסמכים (יווזרים) כאשר לכל יוזר יהיה שם, אימייל וסיסמה

## משימה 2

עדכן את סיסמתו ואת כתובת האימייל של אחד היוזרים שייצרת במשימה 1 לפני ה - pi של אותו היוזר

## משימה 3

מחק את אחד היוזרים שייצרת במשימה 1 לפני ה - pi שלו

## משימה 4

מחק את שאר היוזרים שייצרת במשימה 1 אך השאיר את האוסף (collection) בשם users

# פתרונות לתרגיל

```
use dbtest2
```

תרגיל 1:

```
db.createCollection('users')
```

```
db.users.insertMany([
```

```
    {name: 'Joe', email: 'Joe@gmail.com', password:"123456"},  
    {name: 'Moe', email: 'Moe@gmail.com', password:"1234123!@#"},  
    {name: 'Sacha', email: 'Sacha@gmail.com', password:"!@$!@#$"}  
])
```

תרגיל 2 עדכון לפי ID:

```
db.users.updateOne({_id: ObjectId('656317f683d29d14e6eb2ee4')}, {$set:{email: 'new@gmail.com', password: "newPass123"}})
```

תרגיל 3 מחיקה לפי ID:

```
db.users.deleteOne({_id: ObjectId('656317f683d29d14e6eb2ee4')})
```

תרגיל 4 מחיקה של כל הרשומות - מבלי למחוק את האוסף

```
db.users.deleteMany({})
```

# חיפוש ומציאת של רשומות:

find...

```
db.ads.drop()  
  
db.ads.insertMany(  
[  
  {  
    "title" : "POP show in Tel Aviv",  
    "description": "An article demo for show in tel aviv",  
    "tags": ["pop", "rock", "cool shows"],  
    "showDate": "2016-02-10T10:50:42.389Z",  
    "ticketPrice": 25,  
    "place": "Nokia palace tel aviv"  
  },  
  {  
    "title" : "Shanti banti drums",  
    "description": "Text foo bla bla shanti banti drums",  
    "tags": ["shanti", "drums"],  
    "showDate": "2017-07-23T16:23:42.389Z",  
    "ticketPrice": 10,  
    "place": "Galil area"  
  },  
  {  
    "title" : "Infected mushroom return to the sauce",  
    "description": "Infected mushroom best show ever",  
    "tags": ["psy", "electric", "pop"],  
    "showDate": "2015-09-06T22:00:42.389Z",  
    "ticketPrice": 63,  
    "place": "Haoman 17 jerusalem"  
  },  
  {  
    "title" : "Rock with metallica",  
    "description": "Best rock show in tel aviv",  
    "tags": ["rock", "pop", "metal", "havy metal"],  
    "showDate": "2016-02-17T21:30:42.389Z",  
    "ticketPrice": 52,  
    "place": "Yarkon tel aviv"  
  }  
]  
)
```

יצירת אוסף עבור הדוגמאות הבאות:

# חיפוש ומציאת רשומות: find...

```
> db.ads.find()  
< {  
    _id: ObjectId("65631a0683d29d14e6eb2ee7") ,  
    title: 'POP show in Tel Aviv' ,  
    description: 'An article demo for show in tel aviv' ,  
    tags: [  
        'pop' ,  
        'rock' ,  
        'cool shows'  
    ] ,  
    showDate: '2016-02-10T10:50:42.389Z' ,  
    ticketPrice: 25 ,  
    place: 'Nokia palace tel aviv'  
}  
{  
    _id: ObjectId("65631a0683d29d14e6eb2ee8") ,
```

הציג הרשומות באוסף:  
הmethod find מחזיר cursor  
אובייקט שמאפשר לנו לזרע על כל התוצאות  
באוסף של יותר מ20 רשומות - נשתמש באובייקט  
כדי לראות את יתר התוצאות

# חיפוש ומציאת רשותות:

## findOne

```
> db.ads.findOne()  
< {  
  _id: ObjectId("65631a0683d29d14e6eb2ee7") ,  
  title: 'POP show in Tel Aviv' ,  
  description: 'An article demo for show in tel aviv' ,  
  tags: [  
    'pop' ,  
    'rock' ,  
    'cool shows'  
  ] ,  
  showDate: '2016-02-10T10:50:42.389Z' ,  
  ticketPrice: 25 ,  
  place: 'Nokia palace tel aviv'  
}
```

מציאת רשותות

# חיפוש ומציאה של רשומות:

```
db.ads.find({ticketPrice: 63})
```

↑  
פילטר

```
SELECT * FROM Ads WHERE ticketPrice=63;
```

# שרשור קרייה לפונקציות:

פונקציה שמחזירה מחרוזת:

```
function abc(){return "abc"}
```

בקרייה למתודה abc מקבלים מחרוזת - ולכן - אפשר לשרשר מethodות של מחרוזות.

```
abc().slice(1)
```

# METHODS OF Cursor

```
> db.ads.find().toArray()
```

```
db.ads.find().toArray().length
```

## עוד METHOD של Cursor אחרי הפסקה

# אוסף חדש - עם 999 מסמכים

<https://raw.githubusercontent.com/TomerBu/W160523MR/main/MongoDB/people.json>

```
db.people.insertMany([])
```

```
db.people.find()
```

```
db.people.find().forEach((p)=>{console.log(p)})
```

# איטרציה על מערך:

```
const arr = [1, 2, 3];
```

```
for(let e of arr){  
    console.log(e);  
}
```

```
arr.forEach((e) => {  
    console.log(e);  
});
```

## איטרציה על :Cursor

```
db.people.find().forEach((p)=>{console.log(p)})
```

# חיפוש מסמכים לפי Condition

```
> db.ads.find({"place": "Yarkon tel aviv"})
```



פילטר

# אופרטורים לחיפוש:

## MongoDB Query Operators

```
db.ads.find({ticketPrice: {$lt: 25}})
```

Name	Description	
\$eq	=	Matches values that are equal to a specified value.
\$gt	>	Matches values that are greater than a specified value.
\$gte	>=	Matches values that are greater than or equal to a specified value.
\$in	in	Matches any of the values specified in an array.
\$lt	<	Matches values that are less than a specified value.
\$lte	<=	Matches values that are less than or equal to a specified value.
\$ne	!=	Matches all values that are not equal to a specified value.
\$nin	not in	Matches none of the values specified in an array.

# אופרטורים לחיפוש:

## MongoDB Query Operators

```
db.ads.find({ticketPrice: {$lt: 25}})
```

```
|db.ads.find({ticketPrice: {$lte: 25}})
```

```
› db.ads.find({ticketPrice: {$gt: 25}})
```

```
db.ads.find({ticketPrice: {$eq: 25}})
```

```
› db.ads.find({ticketPrice: {$ne: 25}})
```

# אופרטורים לחיפוש:

\$in \$nin

```
lec2> db.ads.find({ticketPrice: {$in: [25, 63]}})
```

```
db.ads.find({tags: {$in: ['pop', 'rock']}})
```

האופרטור **\$in** מקבל מערך - האם האלמנט תואם לפחות אחד מהאלמנטים בערך

האופרטור **\$nin** מקבל מערך - האם האלמנט לא תואם לפחות אחד מהאלמנטים בערך

```
db.ads.find({ticketPrice: {$nin: [25, 63]}})
```

# באוסף של inventory



```
MongoDB Shell ▾
```

```
db.inventory.insertMany( [ { item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" }, status: "A" }, { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" }, { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status: "A" }, { item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" }, status: "P" }, { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" }, { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" }, { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" }, { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }, { item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" }, status: "A" }, { item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" }, status: "A" } ] );
```

מצאו את האלמנטים שהערך שלהם גדול מ 30

מצאו אלמנטים שהסטטוס שלהם או "A" או "D"

מצאו אלמנטים שהסטטוס שלהם לא "A" או "D"

# פתרונות:



```
MongoDB Shell ▾
```

```
db.inventory.insertMany( [ { item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" }, status: "A" }, { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" }, { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status: "A" }, { item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" }, status: "P" }, { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" }, { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" }, { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" }, { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }, { item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" }, status: "A" }, { item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" }, status: "A" } ] );
```

מצאו את האלמנטים שהערך qty שלהם גדול מ30

```
db.inventory.find({qty: {$gt: 30}}).count()
```

מצאו אלמנטים שהסטטוס שלהם או "A" או "D"

```
db.inventory.find({status: {$in: ["A", "D"]}})
```

מצאו אלמנטים שהסטטוס שלהם לא "A" או "D"

```
db.inventory.find({status: {$nin: ["A", "D"]}})
```

# חיפוש במאגר:

```
db.inventory.deleteMany({});
```

```
db.inventory.insertMany([
  { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
  { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
  { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
  { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
  { item: "postcard", qty: 45, tags: ["blue"] }, dim_cm: [ 10, 15.25 ] }
]);
```

חיפוש כל המוצרים שבסוגtags מהם מכיל red

```
{ db.inventory.find({tags: 'red'}) }
```

## חיפוש בתוך מחרוזת: ביטוי רגולרי: Regex

```
db.users.find({ "company.name": /Dec/ })
```



המחרוזת מכילה

Case Sensitive

```
SELECT * FROM users WHERE 'company.name' LIKE '%Dec%'
```

```
► db.users.find({ "company.name": /llc/i }) /*Case insensitive*/ |
```

המחרוזת מכילה llc

Case Insensitive

## חיפוש בתוך מחרוזת: ביטוי רגולרי: Regex

```
db.users.find({name: /^E/}) /*Starts with E*/
```

המחרוזת מתחילה ב E  
Case Sensitive

```
db.users.find({name: /ell$/}) /*Ends with ell*/
```

המחרוזת מסתיימת ב ell  
Case Sensitive

# אתר אונליין לבדיקה של :Regex

<https://regex101.com/>

The screenshot shows a user interface for testing regular expressions. At the top, a blue header bar contains the text "REGULAR EXPRESSION". Below it is a text input field containing the regular expression pattern: `: / ^Dec`. Underneath this is another blue header bar labeled "TEST STRING". Below it is a text input field containing the string `December`. The entire interface is set against a white background with light gray vertical and horizontal borders.

# מבוא ל:Regex

## Brackets

Brackets are used to find a range of characters:

Expression	Description	
[abc]	abc אחת מהאותיות abc	Find any character between the brackets
[^abc]	abc לא אחת מהאותיות abc	Find any character NOT between the brackets
[0-9]	כל ספרה	Find any character between the brackets (any digit)
[^0-9]	לא ספרה	Find any character NOT between the brackets (any non-digit)
(x y)	או x או y	Find any of the alternatives specified
[xy]	אחד מהאותיות xy	

# מבוא ל:Regex

כל הספרות:

REGULAR EXPRESSION	<code>:/ [0-9]</code>
TEST STRING	123

אותיות גדולות:

REGULAR EXPRESSION	<code>:/ [A-Z]</code>
TEST STRING	abcdtomer123ABC

ספרות או אותיות קטנות:

REGULAR EXPRESSION	<code>:/ [0-9a-z]</code>
TEST STRING	abcdtomer123ABC

הכל מלבד אותיות גדולות:

REGULAR EXPRESSION	<code>:/ [^A-Z]</code>
TEST STRING	abcdtomer123ABC

הכל מלבד ספרות:

[^0-9]

REGULAR EXPRESSION	<code>:/ [abc][oO]</code>
TEST STRING	ooooooooooooeeeeecoo

כל אחת מהאותיות abc  
ואחר כר או O

# Meta Characters

מקוצר את הכתיבה:

## Metacharacters

Metacharacters are characters with a special meaning:

Metacharacter	Description
.	Find a single character, except newline or line terminator
\w	Find a word character
\W	Find a non-word character
\d	Find a digit
\D	Find a non-digit character
\s	Find a whitespace character
\S	Find a non-whitespace character

# Meta Characters

ספרות פי

מקוצר את [0-9]

REGULAR EXPRESSION

```
:/ [\d]
```

TEST STRING

```
12355
```

כל אות למעט ירידת שורה .

REGULAR EXPRESSION

```
:/ .
```

TEST STRING

```
avdc
```

כל אות שיכולה להופיע במילה וו

מקוצר את: [a-zA-Z0-9]

REGULAR EXPRESSION

```
:/ \w
```

TEST STRING

```
any•word•character•including•numbers•[a-zA-Z0-9]
```

כל אות שלא אמורה להופיע במילה וו

מקוצר את: [^a-zA-Z0-9]

REGULAR EXPRESSION

```
:/ \W
```

TEST STRING

```
any•word•character•including•numbers•[a-zA-Z0-9]
```

# Meta Characters

כל דבר שאינו מספר:

REGULAR EXPRESSION

```
:/ \D
```

TEST STRING

```
any•word•character•including•numbers•[a-zA-Z0-9]
```

חיפוש של רווח:

REGULAR EXPRESSION

```
:/ \s
```

TEST STRING

```
any•word•character•including•numbers•[a-zA-Z0-9]
```

לא רווח:

REGULAR EXPRESSION

```
:/ \S
```

TEST STRING

```
any•word•character•including•numbers•[a-zA-Z0-9]
```

# Quantifiers:

ספרה

## Quantifiers

Quantifier	Description
<u>n+</u>	Matches any string that contains at least one <i>n</i>
<u>n*</u>	Matches any string that contains zero or more occurrences of <i>n</i>
<u>n?</u>	Matches any string that contains zero or one occurrences of <i>n</i>
<u>n{X}</u>	Matches any string that contains a sequence of <i>X n's</i>
<u>n{X,Y}</u>	Matches any string that contains a sequence of <i>X to Y n's</i>
<u>n{X,}</u>	Matches any string that contains a sequence of at least <i>X n's</i>

# Quantifiers:

ספרה

רצף של 3 אותיות קטנות

REGULAR EXPRESSION

```
:/ [a-z]{3}
```

TEST STRING

```
abc
```

רצף של 3-5 אותיות קטנות

REGULAR EXPRESSION

```
:/ [a-z]{3,5}
```

TEST STRING

```
abcefa
```

רצף של 0 או יותר אותיות קטנות

REGULAR EXPRESSION

```
[a-z]{0,}
```

TEST STRING

```
12345abcd
```

רצף של 3 תווים ומעלה

REGULAR EXPRESSION

```
:/ [a-z]{3,}
```

TEST STRING

```
abcdefg
```

רצף של אות אחת ומעלה

REGULAR EXPRESSION

```
[a-z]{1,}
```

TEST STRING

```
TOMERabcdef
```

# הרכבה של ביטויי רגולרי:

^ תחילת הביטוי

+ \ ספרה אחת או יותר

- יש מkr

+ \ ספרה אחת או יותר

\$ סיום הביטוי

מספר מקוצר מסטר ^

REGULAR EXPRESSION

```
:/ ^\d+-\d+\$
```

TEST STRING

```
050-123012
```

אUCHO דבר

REGULAR EXPRESSION

```
:/ ^[0-9]+-[0-9]+\$
```

TEST STRING

```
050-123012
```

# הרכבה של ביטוי רגולרי:

\$מספרםילהמספר^

REGULAR EXPRESSION

```
:\ / ^\d+[a-zA-Z]+\d+\$
```

TEST STRING

```
1111aaaa2111|
```

# הרכבה של ביטוי רגולרי:

## REGULAR EXPRESSION

```
: / ^\d{3}-\d{7}\$
```

3 ספרות  
מקר  
7 ספרות

## TEST STRING

```
050-7123012
```

## REGULAR EXPRESSION

```
: / ^0\d{2}-\d{7}\$
```

0  
2 ספרות  
מקר  
7 ספרות

## TEST STRING

```
050-7123012
```

# תרגילים:

```
db.users.find({name: /^b/i})
```

1 מצאו משתמשים שהשם מתחילה ב B או b

```
db.users.find({name: /^[abc]/i})
```

2 מצאו משתמשים שהשם מתחילה ב a או b או c  
(לא חשוב אותיות קטנות או גדולות)

```
db.users.find({email: /^S/})
```

3 מצאו משתמשים שכתובת המail מתחילה באות S

4 מצאו משתמשים שטווס מכיל את המילה apt (CI)

```
db.users.find({"address.suite": /apt/i})
```

5 מצאו companies שיש להם - בשם החברה.  
מקר

```
db.users.find({"company.name": /-/})
```

CI = Case Insensitive  
CS = Case Sensitive

```
db.users.find({"company.name": /\w+-\w+\$/})
```

```
use querytest
```

```
db.ads.insertMany(  
[  
  {  
    "title": "POP show in Tel Aviv",  
    "description": "An article demo for show in tel aviv",  
    "tags": ["pop", "rock", "cool shows"],  
    "showDate": "2023-11-26T14:13:53.933Z",  
    "ticketPrice": 25,  
    "place": "Nokia palace tel aviv"  
  },  
  {  
    "title": "Shanti banti drums",  
    "description": "Text foo bla bla shanti banti drums",  
    "tags": ["shanti", "drums"],  
    "showDate": "2023-11-26T14:13:53.933Z",  
    "ticketPrice": 10,  
    "place": "Galil area"  
  },  
  {  
    "title": "Infected mushroom return to the sauce",  
    "description": "Infected mushroom best show ever",  
    "tags": ["psy", "electric", "pop"],  
    "showDate": "2023-11-26T14:13:53.933Z",  
    "ticketPrice": 63,  
    "place": "Haoman 17 jerusalem"  
  },  
  {  
    "title": "Rock with metallica",  
    "description": "Best rock show in tel aviv",  
    "tags": ["rock", "pop", "metal", "havy metal"],  
    "showDate": "2023-11-26T14:13:53.933Z",  
    "ticketPrice": 52,  
    "place": "Yarkon tel aviv"  
  }  
]
```

# שיעור בית:

← <https://github.com/TomerBu/W160523MR/blob/main/MongoDB/ads.json>

בעזרת המетодה `find` הציגו את כל המסמכים הנמצאים באוסף `ads` שבסיס הנתונים `querytest`

2      כיצד נוכל אם נרצה להביא את המידע מהמסמך השלישי בלבד שנמצא באוסף `ads` (ambilי לדעת מה ה - `id` שלו או כל נתון אחר)?

1

2

# שיעור בית:

- 3 היעזר במתודה find כדי להביא / למשוך רק את המסמכים שמחיר הופעה שלהם (ticketPrice) הוא 10 בלבד
- 4 היעזר במתודה find כדי להביא רק מסמכים שמחירים שווה או גבוהה מ - 52 ושיתקימו במקום Yarkon tel aviv
- 5 היעזר במתודה find כדי להביא רק הופעות שמתויגות כמופע shanti או כמופע electric
- 6 עדכנו את הופעות שמתויגות כונס sahnti שיהיו מתויגותusic relaxing music
- 6 מחקו את הופעות שמתויגות כ-cool shows
- 7 עדכנו את הופעות שמתויגות כ-heavy metal ו-havy metal
- 8 הוחלט לעדכן את כל הופעות למחירים אחידים:
  - עדכנו את מחירי הופעות בין 0 ל-40 כך שמחירן יהיה 35
- 9 מצאו הופעות שמתקיימות בתל אביב
- 10 מצאו הופעות שמתקיימות בירושלים
- 11 מצאו הופעות שבתיאור שלחן מופיע "best" באותיות קטנות או גדולות (CI)
- 12 מצאו הופעות שבתיאור שלחן לא מופיע: "bla" באותיות קטנות או גדולות (CI)