

**Exercise 4: Task 1**

Search: `/ boo / gm`

TEST STRING

```
boo
boomerang
taboo
1000
1000000
0101000010
```

Global pattern flags

- g** modifier: global. All matches (don't return after first match)
- m** modifier: multi line. Causes `^` and `$` to match the begin/end of each line (not only begin/end of string)

MATCH INFORMATION

Match 1	0-3	boo
Match 2	4-7	boo
Match 3	16-19	boo

Boo as literal, selected only text that has “boo” in it.

Search: `/ 1000 / gm`

TEST STRING

```
boo
boomerang
taboo
1000
1000000
0101000010
```

Global pattern flags

- g** modifier: global. All matches (don't return after first match)
- m** modifier: multi line. Causes `^` and `$` to match the begin/end of each line (not only begin/end of string)

MATCH INFORMATION

Match 1	20-24	1000
Match 2	25-29	1000
Match 3	36-40	1000

**Task #2**

1000 as literal, selected 3 strings that have only 100 in them

**REGULAR EXPRESSION** 2 matches (14 steps, 0.0ms)

`/^boo/ gm`

**TEST STRING**

```
boo
boomerang
taboo
1000
1000000
0101000010
```

**EXPLANATION**

- `/^boo/ gm`
- `^` asserts position at start of a line
- `boo` matches the characters `boo` literally (case sensitive)
- Global pattern flags**
  - `g` modifier: global. All matches (don't return after first match)
  - `m` modifier: multi line. Causes `^` and `$` to match the begin/end of each line (not only begin/end of string)

**MATCH INFORMATION**

Match 1	0-3	boo
Match 2	4-7	boo

### Task #3

The **^ Caret** followed by ( `Boo` ) selects only the beginning of a test string.

**REGULAR EXPRESSION** 2 matches (22 steps, 0.1ms)

`/^1000/ gm`

**TEST STRING**

```
boo
boomerang
taboo
1000
1000000
0101000010
```

**EXPLANATION**

- `/^1000/ gm`
- `^` asserts position at start of a line
- `1000` matches the characters `1000` literally (case sensitive)
- Global pattern flags**
  - `g` modifier: global. All matches (don't return after first match)
  - `m` modifier: multi line. Causes `^` and `$` to match the begin/end of each line (not only begin/end of string)

**MATCH INFORMATION**

Match 1	20-24	1000
Match 2	25-29	1000

### Task #4

Same as task 3

REGULAR EXPRESSION

1 match (14 steps, 0.0ms)

/ ^boo\$ / gm

TEST STRING

boo  
boomerang  
taboo  
1000  
1000000  
0101000010

EXPLANATION

/ ^boo\$ / gm

^ asserts position at start of a line

boo matches the characters boo literally (case sensitive)

\$ asserts position at the end of a line

**Global pattern flags**

g modifier: global. All matches (don't return after first match)

m modifier: multi line. Causes ^ and \$ to match the begin/end of each line (not only

MATCH INFORMATION

Match 1

0-3

boo

## Task #5

The \$ symbol states that only at the end of a string should (boo) match, also ^ states only the beginning should match

REGULAR EXPRESSION

1 match (22 steps, 0.1ms)

/ ^1000\$

TEST STRING

boo  
boomerang  
taboo  
1000  
1000000  
0101000010

EXPLANATION

/ ^1000\$ / gm

^ asserts position at start of a line

1000 matches the characters 1000 literally (case sensitive)

\$ asserts position at the end of a line

**Global pattern flags**

g modifier: global. All matches (don't return after first match)

m modifier: multi line. Causes ^ and \$ to match the begin/end of each line (not only

MATCH INFORMATION

Match 1

20-24

1000

## Task #6

Same as task 5

### Exercise 5: Task #1

REGULAR EXPRESSION

1 match (36 steps, 0.1ms)

⋮

/

`^\d\d\s\w\w\w\w\w\w\s\w\w\w\w\w\w\d\s\w\w\w\w`

/gm

`\w\w\d\s\w\w\w\w\s\w\w\w`

TEST STRING

`22*Acadia*Avvenue,*London,*East*End`

EXPLANATION

▼

`/^\d\d\s\w\w\w\w\w\w\s\w\w\w\w\w\w\d\s\w\w\w\w`

/gm

`^` asserts position at start of a line

`\d` matches a digit (equivalent to `[0-9]`)

`\d` matches a digit (equivalent to `[0-9]`)

`\s` matches any whitespace character (equivalent to `[\r\n\t\f\v+]`)

`\w` matches any word character (equivalent to `[a-zA-Z0-9_]`)

MATCH INFORMATION

Match 1

0-34

`22*Acadia*Avvenue,*London,*East*End`

### Exercise 6 Task#1

REGULAR EXPRESSION

1 match (39 steps, 0.1ms)

/

\D\D\D\D\D\D\D\D\D\D\D\D,

\D\D\D\D\D\D\D,

\D

/ gm

TEST STRING

John\*Q.\*Adams,\*Denver,\*Colorado,\*80123

EXPLANATION

/

\D\D\D\D\D\D\D\D\D\D\D\D,

\D

/ gm

\D matches any character that's not a digit (equivalent to `[^0-9]`)

\D matches any character that's not a digit (equivalent to `[^0-9]`)

\D matches any character that's not a digit (equivalent to `[^0-9]`)

\D matches any character that's not a digit

MATCH INFORMATION

Match 1

0-38

John\*Q.\*Adams,\*Denver,\*Colorado,\*80123

## Exercise 7

REGULAR EXPRESSION 2 matches (16 steps, 0.1ms)

:/ Flavou?r / gm

TEST STRING

Flavour  
Flavor

EXPLANATION

- Flavou?r / gm
  - Flavo matches the characters Flavo literally (case sensitive)
  - u matches the character u with index 117<sub>10</sub> (75<sub>16</sub> or 165<sub>8</sub>) literally (case sensitive)
  - ? matches the previous token between zero and one times, as many times as possible, giving back as needed (greedy)
  - r matches the character r with index 114<sub>10</sub> (72<sub>16</sub> or 162<sub>8</sub>) literally (case sensitive)

MATCH INFORMATION

Match 1	0-7	Flavour
Match 2	8-14	Flavor

?? Typing the literal word followed by a ? selects both test strings, but doesn't give you one match.

## Exercise 8

REGULAR EXPRESSION 1 match (11 steps, 0.0ms)

:/ \D{5}\s\D{5}\s\D{8}\s\D{7}\s\D{11}\D / gm

TEST STRING

Today•we're•learning•regular•expressions.

EXPLANATION

- \D{5}\s\D{5}\s\D{8}\s\D{7}\s\D{11}\D / gm
  - \D matches any character that's not a digit (equivalent to `[^0-9]`)
  - {5} matches the previous token exactly 5 times
  - \s matches any whitespace character (equivalent to `[\r\n\t\f\v+]`)
  - \D matches any character that's not a digit (equivalent to `[^0-9]`)

MATCH INFORMATION

Match 1	0-41	Today•we're•learning•regular•expressions.
---------	------	---

## Exercise 9

REGULAR EXPRESSION
1 match (52 steps, 0.1ms)

```

:: / \w{5}\sis\s\w{3}\s\w{4}\s\w\w\s\w{3}\sday\n\w / gm
{6}\sis\s\w{3}\s\w{4}\s\w\w\s\w{3}\sday\nis\s
\w{3}\s\w{4}\s\w\w\s\w{3}\sday

```

TEST STRING

```

Apple=is=the=word=of=the=day
Banana=is=the=word=of=the=day
is=the=word=of=the=day

```

EXPLANATION

- ▼ / \w{5}\sis\s\w{3}\s\w{4}\s\w\w / gm
  - ▼ \w matches any word character (equivalent to [a-zA-Z0-9\_])
  - {5} matches the previous token exactly 5 times
  - \s matches any whitespace character (equivalent to [\r\n\t\f\v])

MATCH INFORMATION

Match	Index	Text
Match 1	0-81	Apple=is=the=word=of=the=day Banana=is=the=word=of=the=day is=the=word=of=the=day

## Exercise 10

REGULAR EXPRESSION
1 match (7 steps, 0.0ms)

```

:: / \D{18}\n\D{12}\D{17}\n\D{28} / gm

```

TEST STRING

```

User=Tom=logged=in
User=contoso\batman=logged=in
User=contoso\joker=logged=in

```

EXPLANATION

- ▼ / \D{18}\n\D{12}\D{17}\n\D{28} / gm
  - ▼ \D matches any character that's not a digit (equivalent to [^0-9])
  - {18} matches the previous token exactly 18 times
  - \n matches a line-feed (newline) character (ASCII 10)
  - ▼ \D matches any character that's not a digit (equivalent to [^0-9])
  - {12} matches the previous token exactly 12

MATCH INFORMATION

Match	Index	Text
Match 1	0-77	User=Tom=logged=in User=contoso\batman=logged=in User=contoso\joker=logged=in

## Exercise 11 Task 1

The screenshot shows the Regex 101 interface. The regular expression is `/gr[ae]y/gm`. The test string contains `gray` and `grey`. The explanation states that `gr` matches the characters `gr` literally, `[ae]` matches a single character in the list `ae`, and `y` matches the character `y`. The match information shows two matches: Match 1 (0-4) for `gray` and Match 2 (5-9) for `grey`.

## Exercise 11 Task 2

Used an 'A' as it is literal for the Reg Ex

The screenshot shows the Regex 101 interface. The regular expression is `/Advis[eo]r/gm`. The test string contains `Adviser` and `Advisor`. The explanation states that `Advis` matches the characters `Advis` literally, `[eo]` matches a single character in the list `eo`, and `r` matches the character `r`. The match information shows two matches: Match 1 (0-7) for `Adviser` and Match 2 (8-15) for `Advisor`.

## Exercise 11 Task 3

The screenshot shows the Regex 101 interface. The regular expression is `/a?esthetic/gm`. The test string contains `aesthetic` and `esthetic`. The explanation states that `a` matches the character `a`, `esthetic` matches the characters `esthetic` literally, and the `g` modifier is global. The match information shows one match: Match 1 (0-10) for `aesthetic`.

## Exercise 12

regular expressions 101

social

donate

info

REGULAR EXPRESSION

5 matches (20 steps, 0.0ms)

/ 0x[0-9,a-f,A-F]+ / gm

TEST STRING

0x000000  
0xFFFFF  
0x000000  
0x0000DF  
0x000A00

EXPLANATION

▼ / 0x[0-9,a-f,A-F]+ / gm

► 0x matches the characters 0x literally (case sensitive)

▼ Match a single character present in the list below [0-9,a-f,A-F]

+ matches the previous token between one and unlimited times, as many times as possible, giving back as needed (greedy)

0-9 matches a single character in the

MATCH INFORMATION

Match 1 0-8 0x000000

Match 2 9-17 0xFFFFF

## Exercise 13

```

C:\Users\washa>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : DESKTOP-SDE78TA
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    DNS Suffix Search List. . . . . : hsd1.fl.comcast.net

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . : hsd1.fl.comcast.net
    Description . . . . . : Intel(R) I211 Gigabit Network Connection
    Physical Address. . . . . : 70-4D-7B-8B-B0-93
  
```