

# Python

Python is a high level, interpreted & general purpose, dynamic programming language which focus on used in many fields.

Human  
Readable

```
print ("Hello");
```

C/C++  
cout << "\_\_\_"  
Machine

CODE || READABILITY

## Who's Using Python?

- \* Software Engineers
- \* Web Developers.
- \* Data Scientists
- \* Network Engineers
- \* Data Analysts
- \* Mathematicians
- \* Accountants

Kids

## Why Python?

① Easy of Coding → learning curve of Python is very low.

```
str="Hello World"
```

Solve complex problems using less time & with fewer lines of code

Java

```
str.substring(0,3)
```

```
str.substring(0,3)
```

JS

```
str[0:3]
```

Python

```
str[0:3]
```

## ② GENERAL PURPOSE

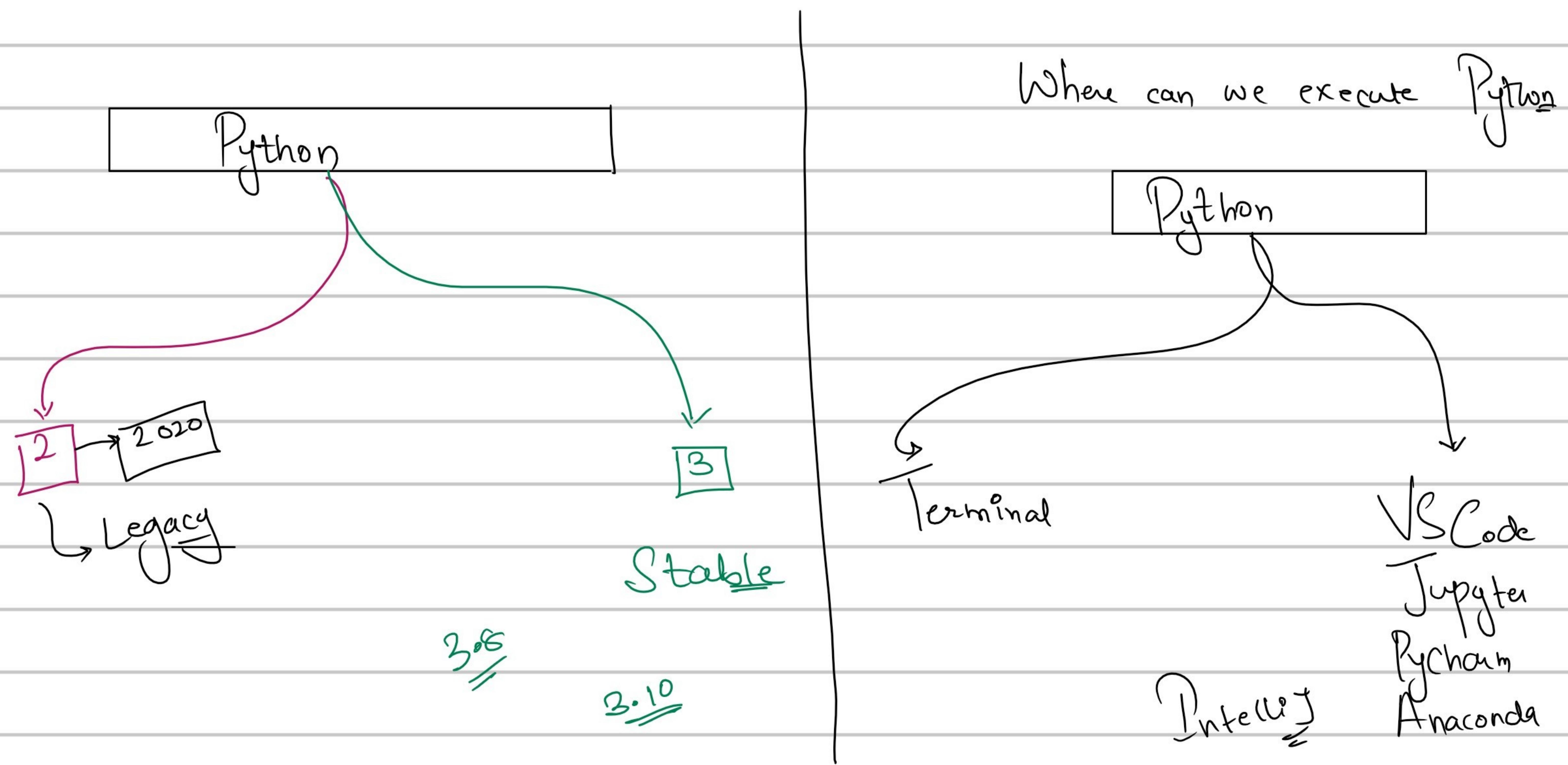
- ① Data Science
- ⑥ Testing
- ⑪ Desktop App
- ② Data Analytics
- ⑤ AI
- ⑫ Video Games
- ③ Machine Learning
- ⑧ Big Data
- ⑬ Data Visualization
- ④ Trading Programs
- ⑨ Mobile App
- ⑮ Statistics
- ⑦ Automation.
- ⑯ API Development

## ② High Level

- ⑦ Cross Platform → no dependency on OS

⑤ Large Ecosystem → there are tons of libraries available for python.

## ⑥ Huge Community Support



# Data Types in Python

Type	Data Types
Text	str
Numeric	int, float, complex
Sequence	list, range, tuple
Map	dict
Set	set, frozenset
Boolean	bool
Binary	byte, bytearray, memoryview

## Keywords in Python

Keyword	Description
and //	A logical operator
as //	To create an alias
assert //	For debugging
break ,	To break out of a loop
class //	To define a class
continue ,	To continue to the next iteration of a loop
def //	To define a function
del //	To delete an object
elif ↗ else ↗ if	Used in conditional statements, same as else if
else	Used in conditional statements
except ↗ catch //	Used with exceptions, what to do when an exception occurs

## Keywords in Python

Keywords	Description
false	Boolean value, result of comparison operations
False	Boolean value, result of comparison operations
finally //	Used with exceptions, a block of code that will be executed no matter if there is an exception or not
for	To create a for loop
from	To import specific parts of a module
global ↗	To declare a global variable
if ↗	To make a conditional statement
import ↗	To import a module
in ↗	To check if a value is present in a list, tuple, etc.
is ↗	To test if two variables are equal
lambda ↗	To create an anonymous function

## Keywords in Python

Keyword	Description
None //	Represents a null value
nonlocal ,	To declare a non-local variable
not //	A logical operator
or //	A logical operator
pass //	A null statement, a statement that will do nothing
raise ,	To raise an exception
return //	To exit a function and return a value
True ↗	Boolean value, result of comparison operations
try //	To make a try...except statement
while //	To create a while loop
with //	Used to simplify exception handling
yield //	To end a function, returns a generator

# Conventions in Python

\* function | variable

get\_location\_for\_car()  
tax\_amount

Snake Casing →

One\_two\_three-four

\* class

Pascal Casing

OneTwoThreefour

BankDetail

MyPoint

1

1 2 ③ 4 5 6

range

[ from , to ]

↓

↓

k i t k a t

zero based

one

based

0 →

0 ① 2 3 4 5

-ve

-6 -5 -4 -3 -2 -1

word[1 : 3]

1 1 2 3 4 ⑤ 6 7 8 9 10 ⑪ 12 ⑬ 14 15 16 17 18 19 20  
I n t e r n a t i o n a l i z a t i o n  
0 ① 1 2 3 4 ⑥ 6 7 ⑧ 9 10 11 12 13 14 15 16 17 18 19

international 0:13  
inter 0:5  
intan 0:6

nation 5:11  
national 5:13  
ion 8:11, 17: 9:11

on 16:20  
18:1

$\rightarrow$  space

## Flow Control

{  
g X

Indentation

if condition:  
 $\star \star \star \star \star$  statements

else

$\star \star \star \star \star$  statements

String Interpolation in Python

a=10  
b=20

When we write a string  
without breaking for

including variables

[ "Value of a is "+a+", and value of b is "+b]

message=f" Value of a is {a}, value of b is {b}";

# Functions

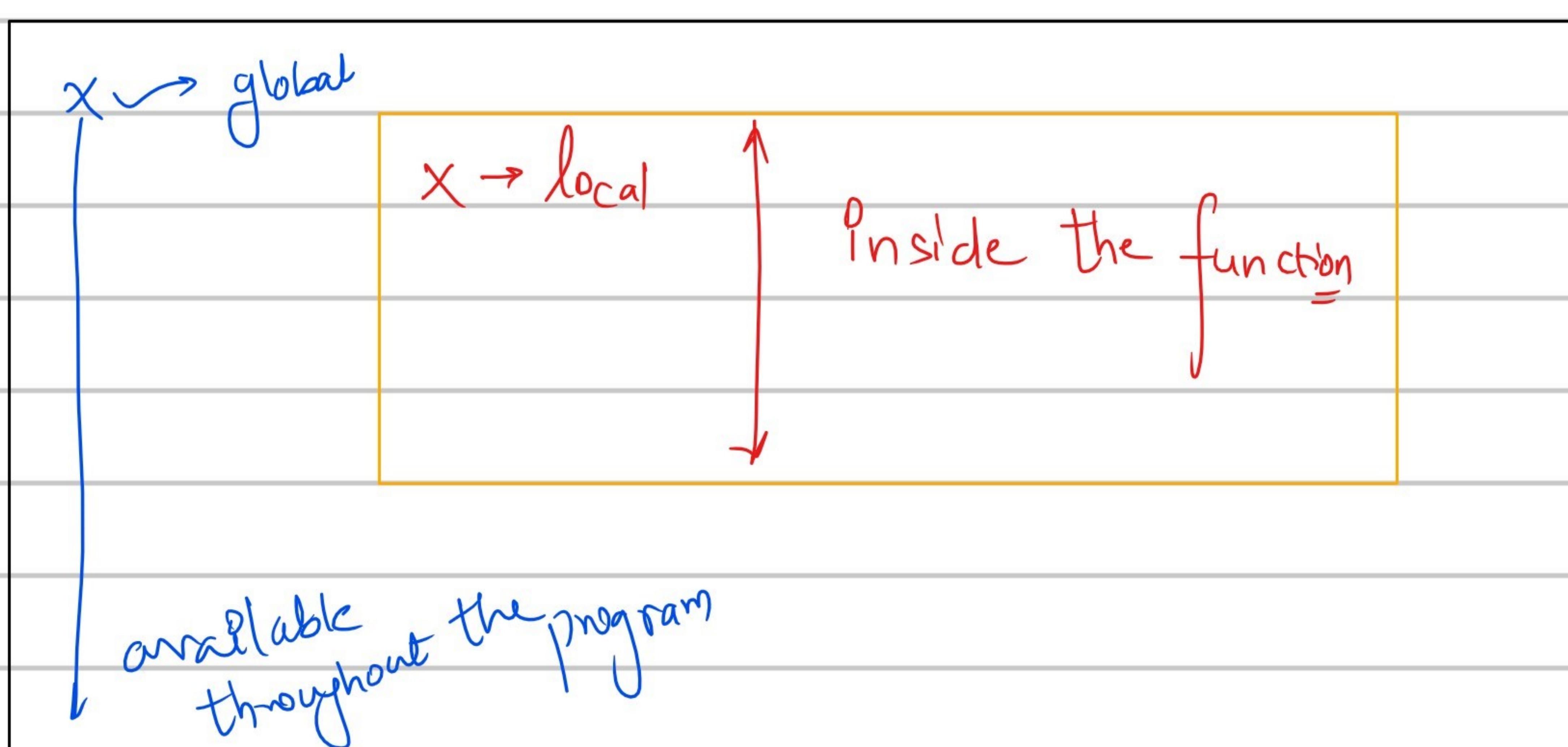
group of related statements  
which performe a  
specific task.

CODE ON DEMAND

```
def function_name(parameters):  
    """ doc_string """  
    statement(s)  
    return someValue,
```

→ Optional

## Scopes in Python



\* Python doesn't support block scope  
\* min scope it supports is function

# Data Structures

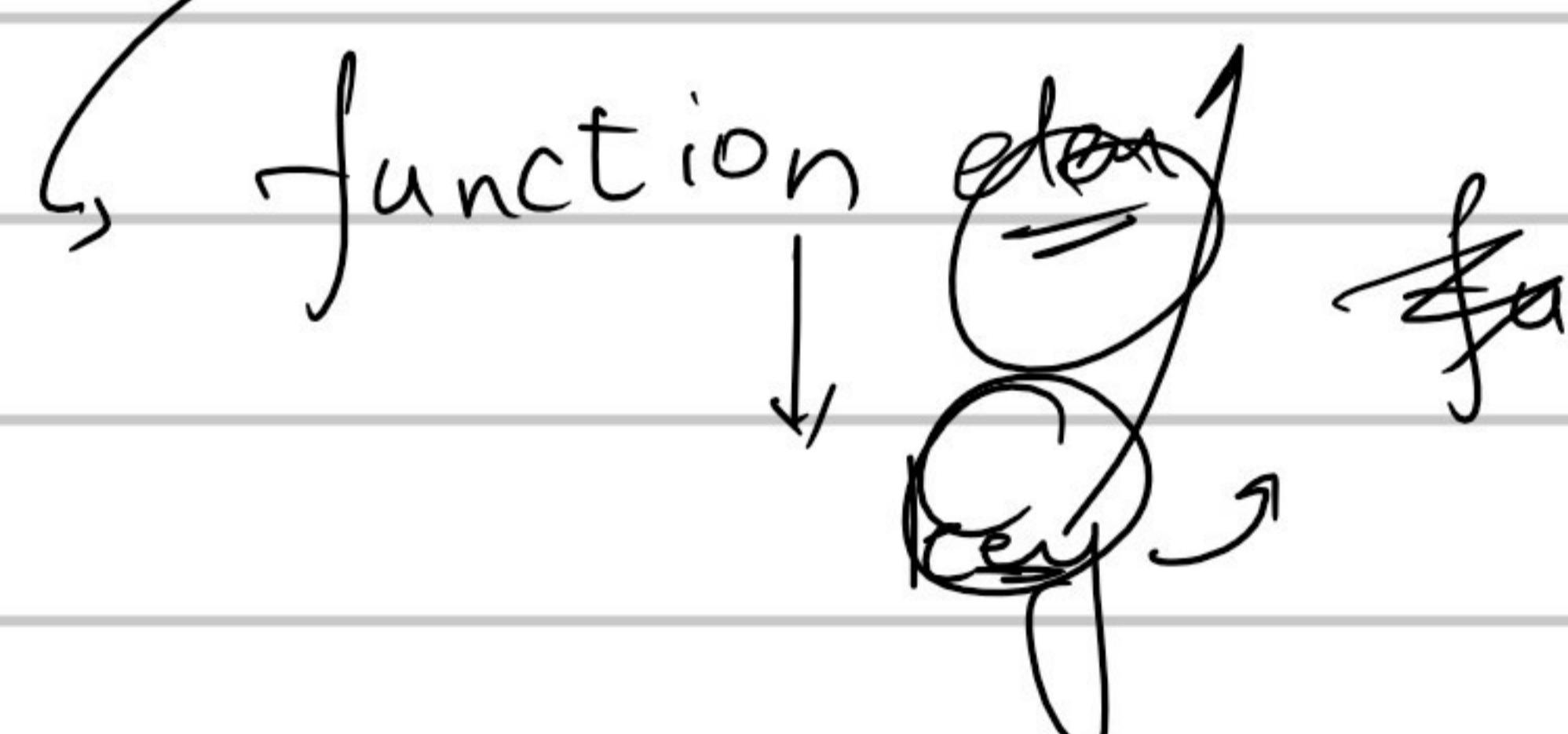
list tuple range set dict

# Sorting

numbers = [ 1, 5, 8, 0, 9 ]

[0, 1, \mathbb{S}, \mathbb{E}, 9]

```
Cart = [ (prod, price),  
        (prod, price),  
        (prod, price) ]
```



# Lambda Functions

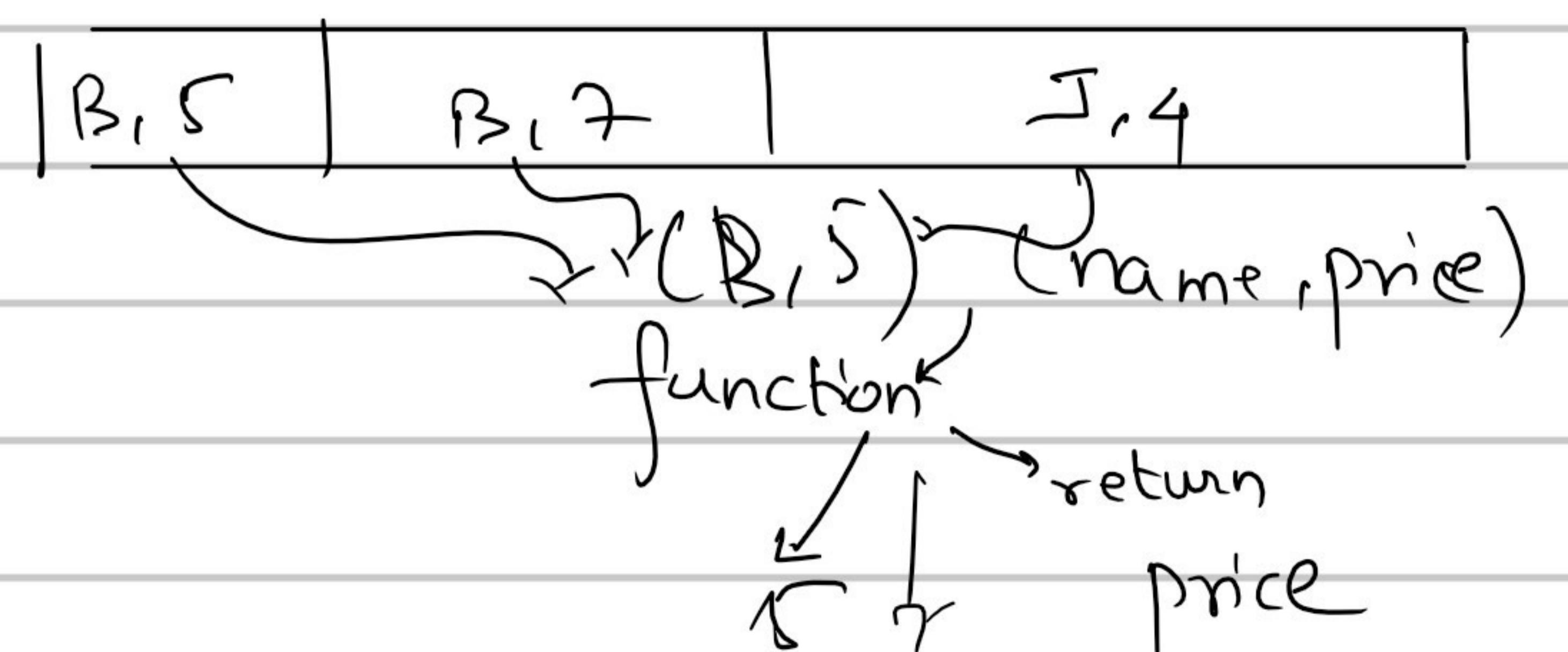
→ anonymous functions → functions without any name.

# Structure of a lambda function

lambda Parameter : expression

no return statement  
expression → executable  
↓ execute  
else  
↳ value  
↓  
return

map.



S	7	4
---	---	---

## Exception handling

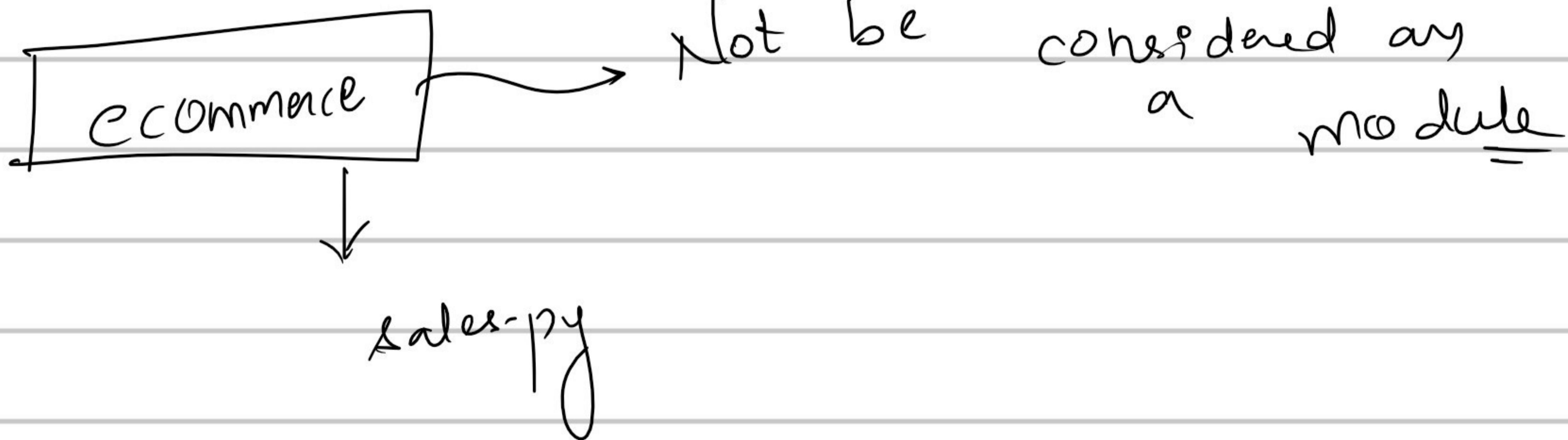
→ unexpected erroneous event which breaks the execution of the code.

try:

except Error:

## Modules

allows for the segregation in the code.



\* Regex \*

Regular Expression

zartab@codewithz.com ✓

zartab@abc@xyz.com ✗

zartab0@gmail.com ✗

finding | matching

Pattern in a  
string

Regex

Basic

Extended

Symbol	What it represents?
*	Zero or more occurrences of the character that preceeds this asterisk
.	A wild card that represents any character <i>(alphanumeric)</i>
\s	Represents whitespace
[pqr]	A single character which can be either a 'p' , 'q' or an 'r'
[a-d]	A single character that falls in the range 'a-d'
[^pq]	A single character that is neither 'p' nor 'q'
^pattern	^ is an anchor tag that represents the beginning of line
pattern\$	\$ is an anchor tag that represents the end of line



Understand the Requirement: What needs to be included or excluded

- fooaaaabar
- fooabar
- foobar
- fooaabbar
- fooxxxbar
- fooxbar

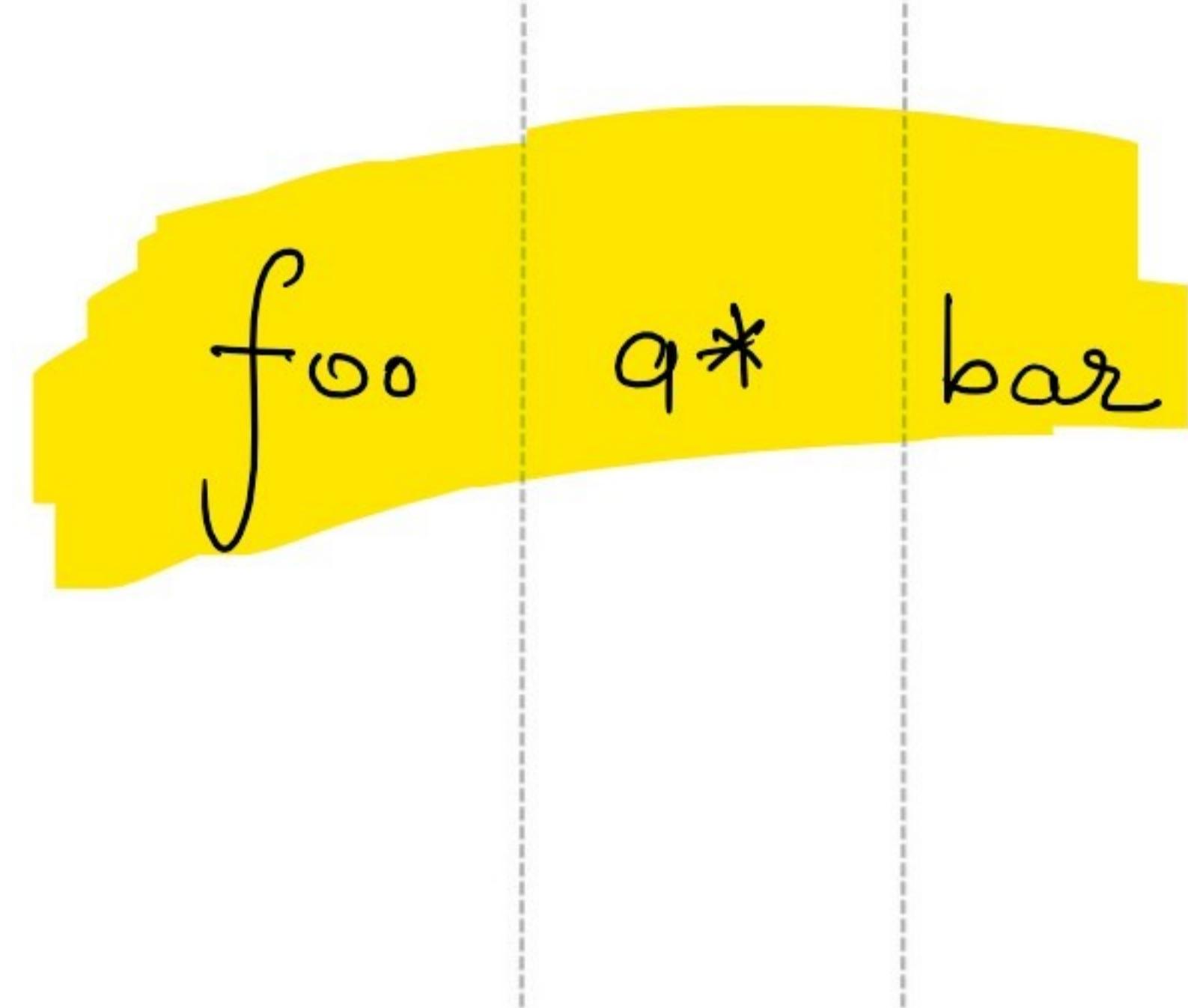


Identify the pattern in inclusion or exclusion list

- foo aaaa bar
- foo a bar
- foo aa bar



Final Regular Expression



**a\* - Zero or more occurrences o 'a' (The character just preceding the asterisk)**



Understand the Requirement: What needs to be included or excluded

- foo3bar
- fooabar
- fooxbar
- baryfoo
- foobar
- foooxybar
- foocbar

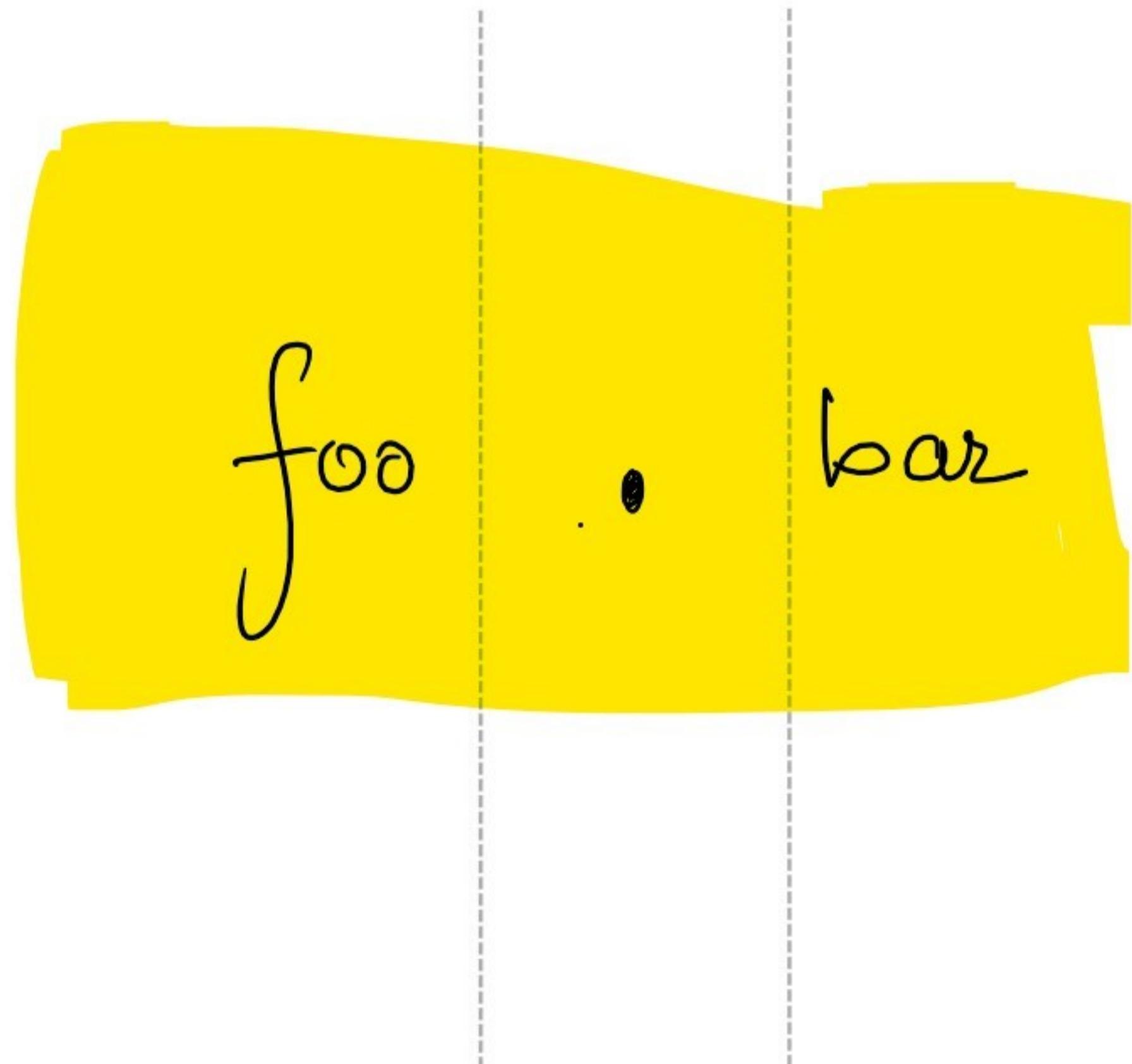


Identify the pattern in inclusion or exclusion list

- foo (a) bar
- foo (x) bar
- foo (c) bar
- foo (.) bar



Final Regular Expression



**. - Single wildcard . Can represent only ONE character [any] in single position**

1

Understand the Requirement: What needs to be included or excluded

- foobar
- barfoo
- fooabcbbar
- fooxcbar
- barcbyfoo
- foozbar
- barafoo
- barabfoo

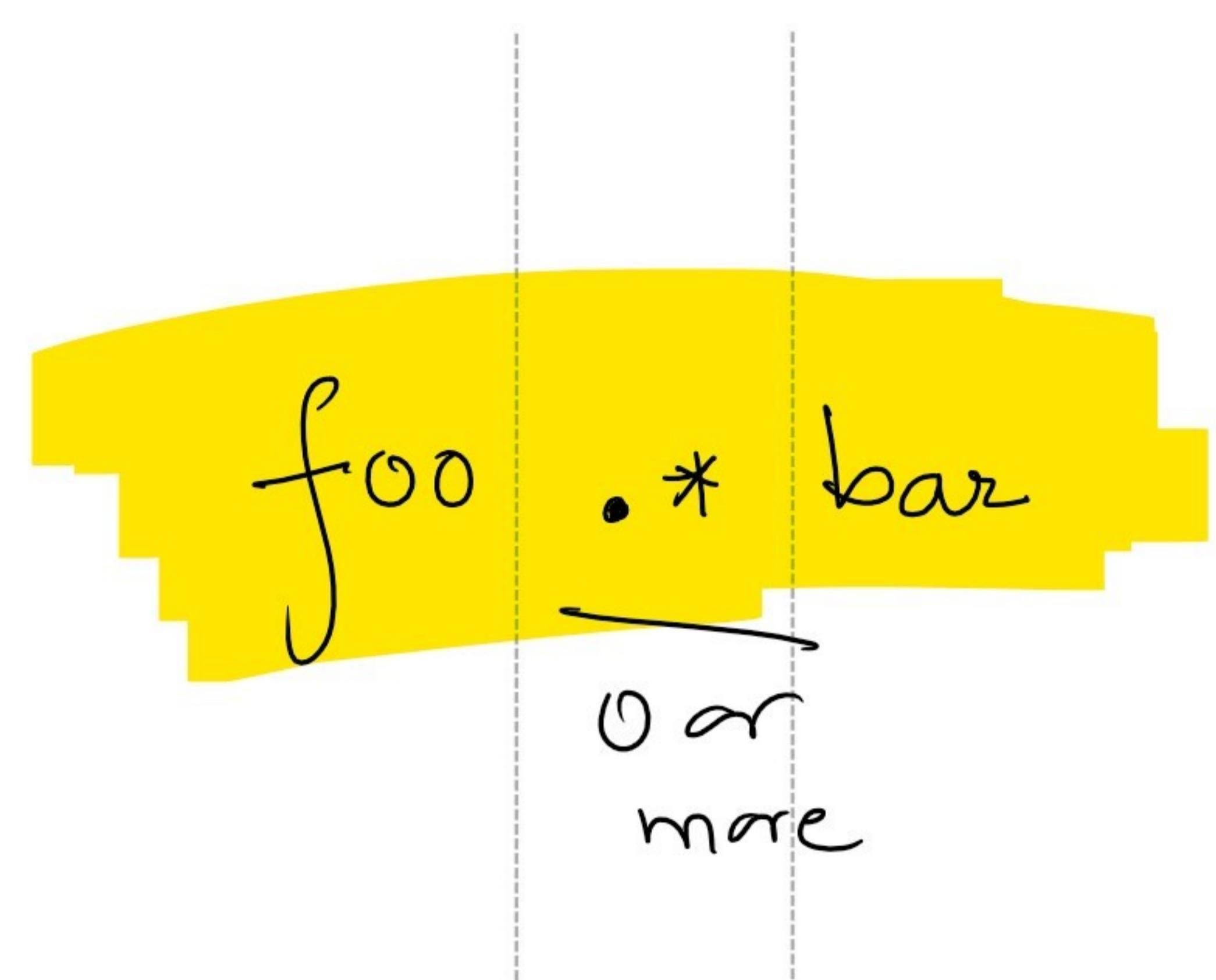
2

Identify the pattern in inclusion or exclusion list

<input checked="" type="checkbox"/> foo	_o	bar
<input checked="" type="checkbox"/> foo	abc	bar
<input checked="" type="checkbox"/> foo	xc	bar
<input checked="" type="checkbox"/> foo	z	bar

3

Final Regular Expression



.\* - Zero or more occurrences of wildcard, which means zero or more occurrences of any character

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

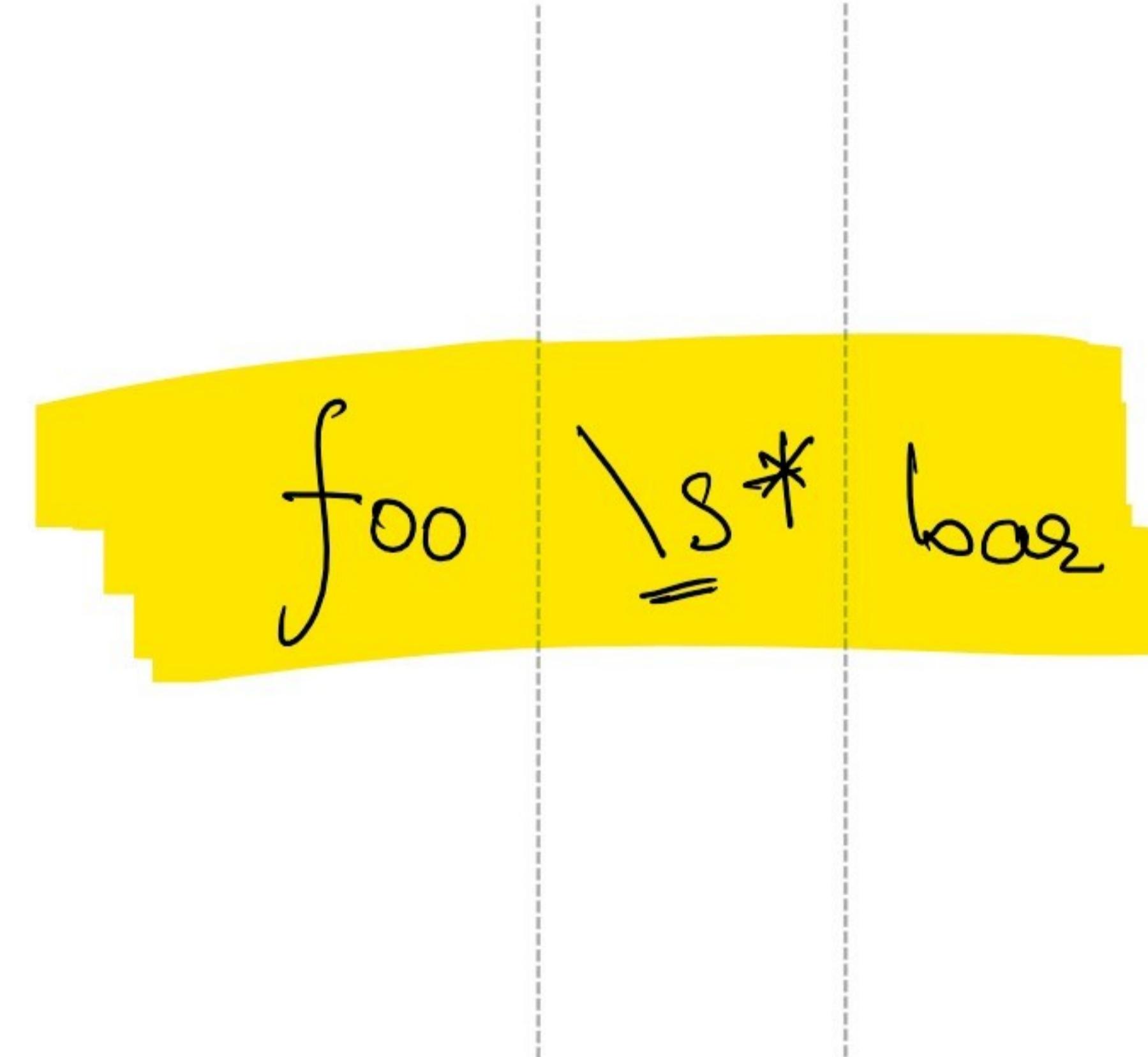
3

Final Regular Expression

\s → space

- fooxxxbar
- foo bar
- foobar
- fooxxbbar
- foo bar
- foo bar
- foobar
- fooyyybar

<input checked="" type="checkbox"/> foo	<3 spaces>	bar
<input checked="" type="checkbox"/> foo	<1 spaces>	bar
<input checked="" type="checkbox"/> foo	<6 spaces>	bar
<input checked="" type="checkbox"/> foo	<0 spaces>	bar



\s represents whitespace. \s\* represents zero or more occurrence of whitespace

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

foo  
 moo  
 coo  
 doo  
 poo  
 loo  
 boo  
 hoo

f oo  
 c oo  
 l oo

either one of the options

[fcl] oo

**[abc] - Character class. One of the character inside the square brackets - a,b or c**

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

foo  
 moo  
 coo  
 doo  
 poo  
 loo  
 boo  
 hoo

m oo  
 h oo

[fcndl] oo

[^mh] oo

**[^abc] - Any character EXCEPT any of the one inside the square brackets, in a single**

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

- joo
- moo
- koo
- loo
- poo
- zoo
- moo
- hoo

- j oo
- k oo
- l oo
- m oo

$[jklm]$  oo

$[j-m]$  oo

### [a-c] - One of the character falling in the range given in square brackets

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

- joo
- moo
- koo
- loo
- poo
- moo
- zoo
- hoo

- j oo
- k oo
- l oo
- m oo
- z oo

anything in j,k,l,m & z

$[j-mz]$  oo

### [a-cx] - One of the character falling in the range given in square brackets OR any of the other choices given in square brackets - a,b, c,x

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

- joo
- moo
- Koo
- Loo
- poo
- moo
- zoo
- hoo

<input checked="" type="checkbox"/> j	oo
<input checked="" type="checkbox"/> K	oo
<input checked="" type="checkbox"/> L	oo
<input checked="" type="checkbox"/> m	oo
<input checked="" type="checkbox"/> z	oo

Anything in j, k, l, m, J, K, L, M  $\Sigma_j$

$[j-mJ-Mz] \Sigma_0$

**[a-cACx]** - One of the character falling in the range given in square brackets OR any of the other choices given in square brackets - a,b, c,A,B,C,x

1

Understand the Requirement: What needs to be included or excluded

yx  
 yxxx

xxxx.yyy

xx.yyyy

x.yy

xy

xx yy

yy xx

2

Identify the pattern in inclusion or exclusion list

xxx	.	yy
xx	.	yyy
x	.	yy
	.	

3

Final Regular Expression

don't use it as a wildcard qt as a period  $\Sigma_{\text{any}}$

$x^* \backslash \cdot y^*$

If a '.' is inside square brackets, it need not to be escaped

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

x#y

x:y

x.y

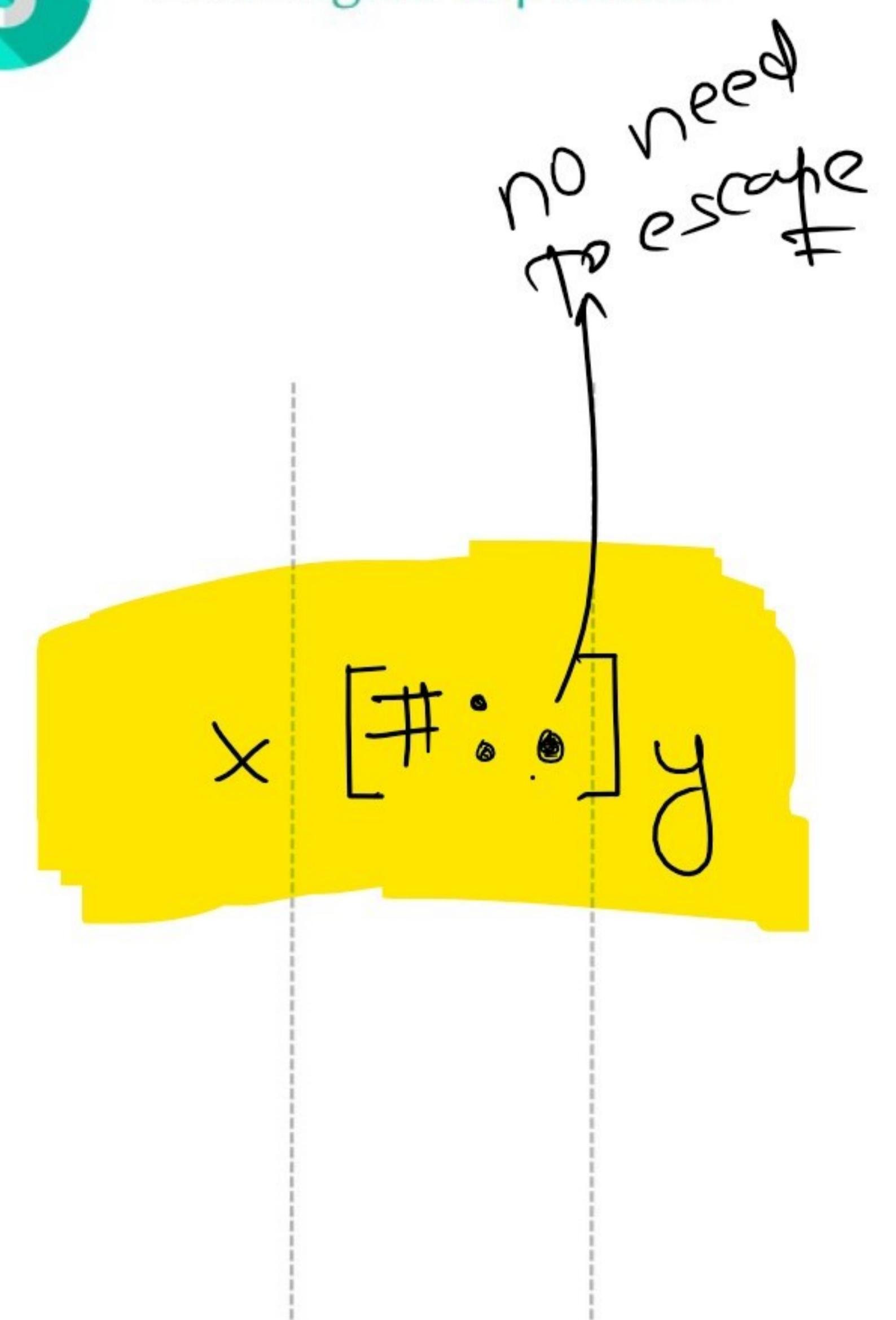
x&y

x%y

x # y

x : y

x . y



If a '.' is inside square brackets, it need not to be escaped

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

x#y

x:y

x^y

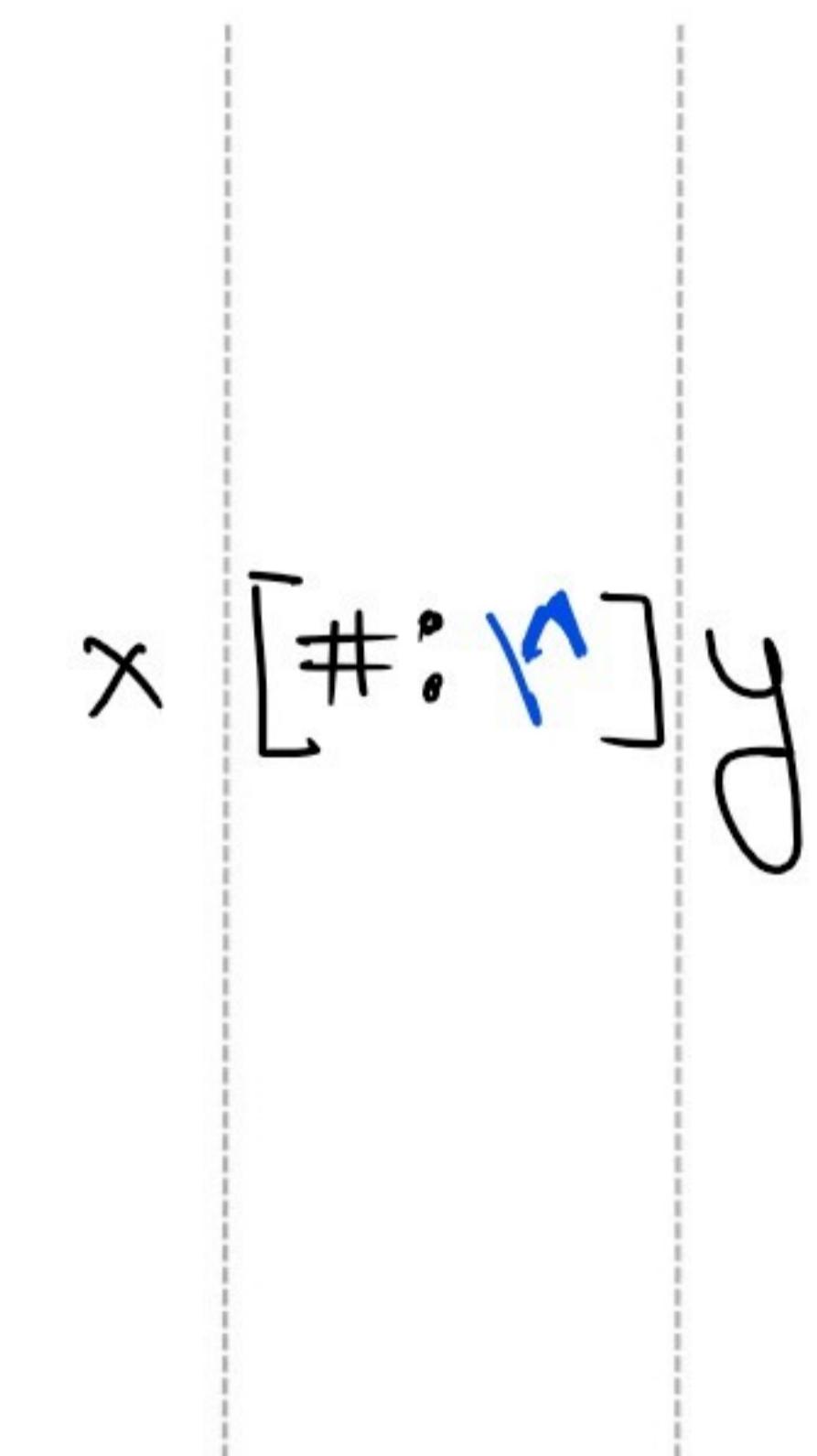
x&y

x%y

x # y

x : y

x ^ y



If a '^ - ' is inside square brackets, it need to be escaped with backslash

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

x#y

x # y

x\y

x \ y

x^y

x ^ y

x&y

x%y

x [ # \| ^ ] y

If a '\ ' is inside square brackets, it need to be escaped with another backslash

































































