

Python

low learning curve

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

which focus on

cout << "Hello";
print ("Hello"); ~ High Level

CODE READABILITY

Who's using Python?

* Software Engineers

* Web Developers

/ * Data Analysts

* Network Engineers

* Data Scientists

* Search Engines

* Mathematicians / Accountants

Kid's =

Why Python?

① Ease of Coding

learning curve of Python is very low

str = "Hello World"

Solve complex problems
using less lines & fewer lines
of code

Java

str.substring(0,3)

JS

str.substring(0,3)

Python

str[0:3]

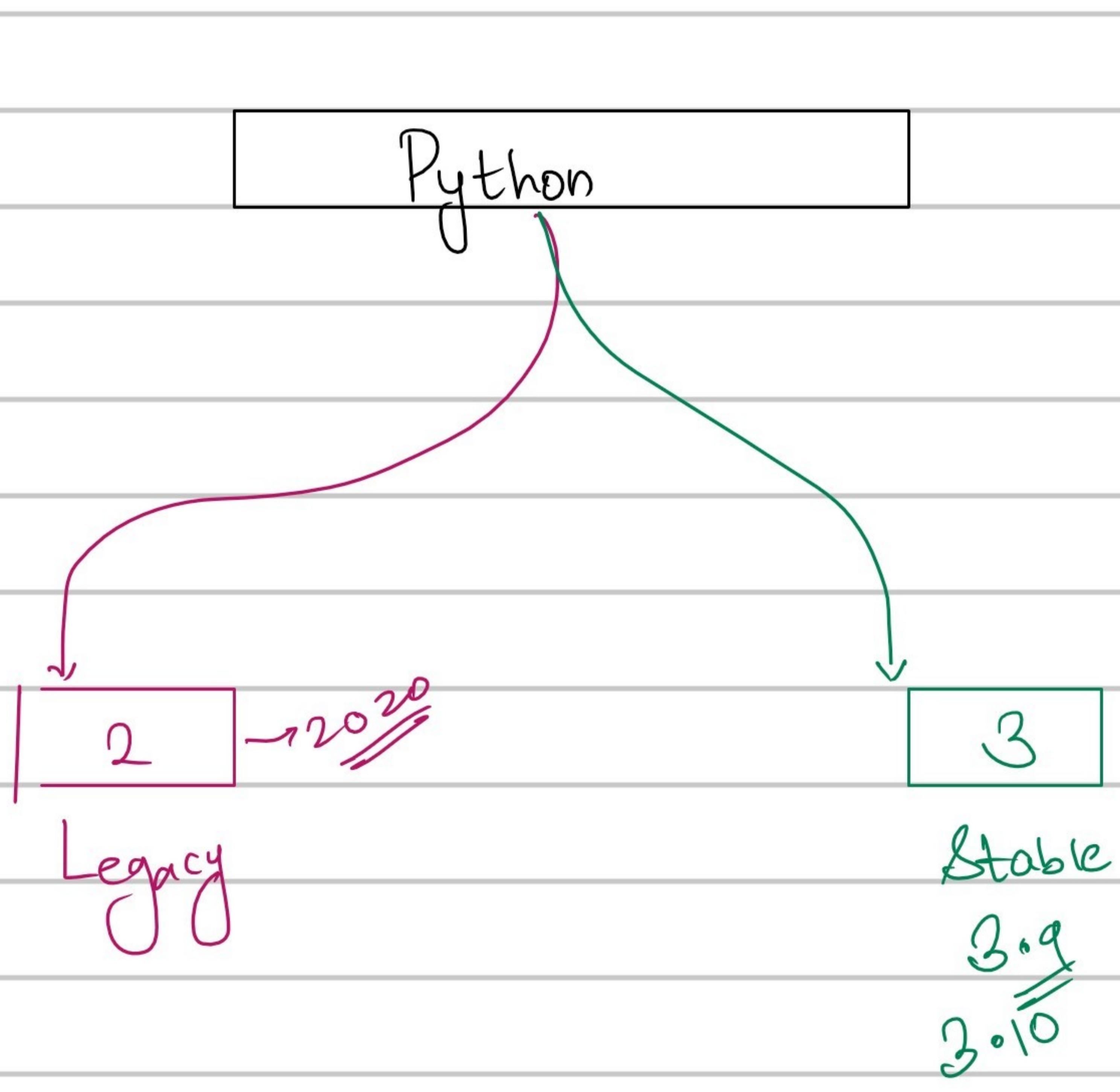
② General Purpose

- ① Data Science
- ② Data Analytics
- ③ Machine Learning
- ④ Trading Programs
- ⑤ Automation
- ⑥ Testing
- ⑦ AI
- ⑧ Big Data
- ⑨ Mobile App
- ⑩ Desktop Application
- ⑪ Video Games
- ⑫ Data Visualization
- ⑬ Statistics
- ⑭ Web Development
- ⑮ API Development
- ⑯ High Level

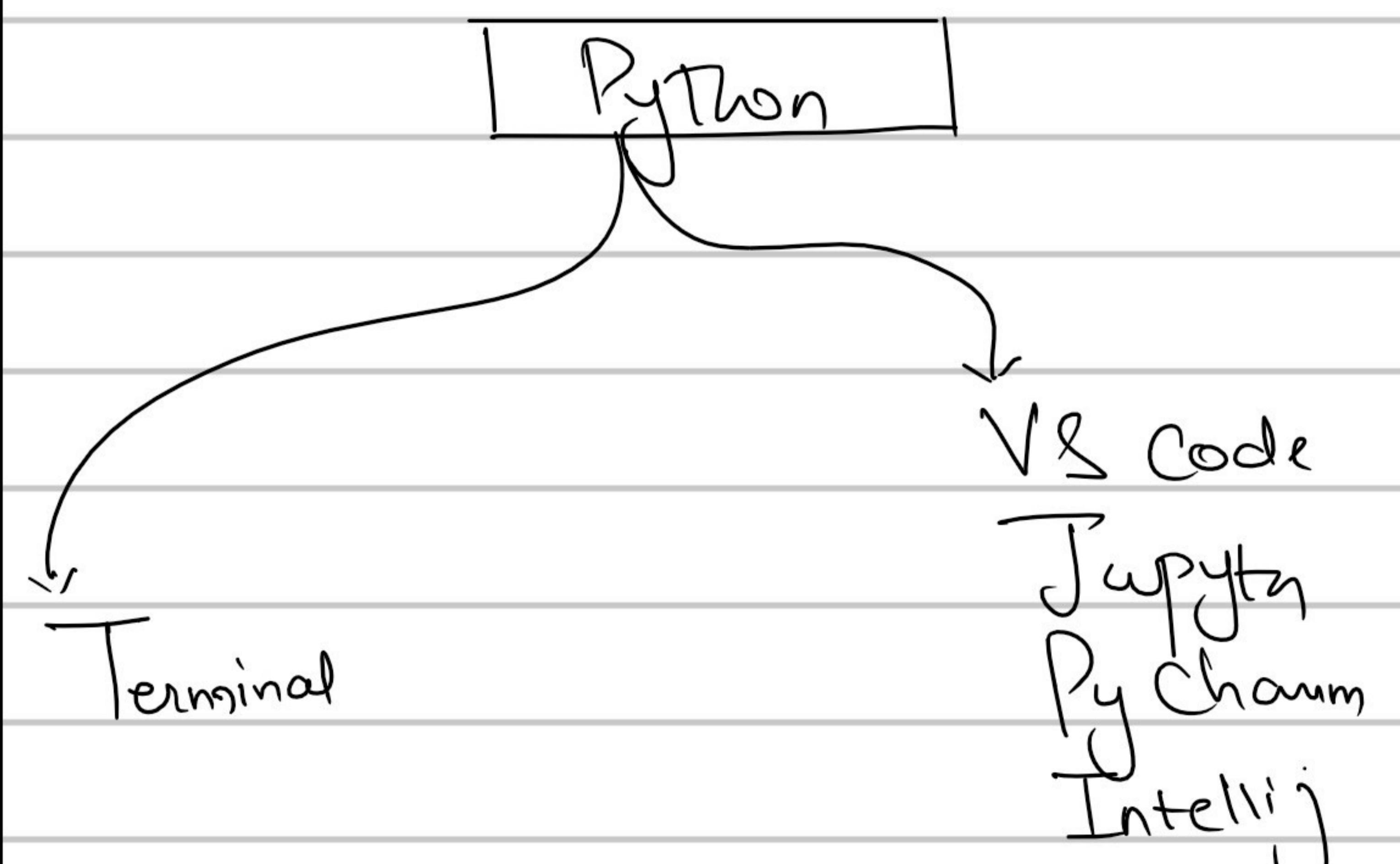
④ Cross Platform → no dependency on OS

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

⑥ Huge Community Support



Where do we execute Python



Data Types in Python

Type
Text
Numeric
Sequence
Map
Set
Boolean
Binary

Data Types

str
int, float, complex
list, range, tuple
dict
set, frozenset
bool
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 ↗ 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 ↗ numbers	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

* functions | variables

tax_amount

get_location_for_car()

Snake Casing → one_two_three_fours

*

class

BankDetail

LocationManager

Pascal Casing → OneTwoThreeFour

range

0 →

0 1 2 3 4 5

[from : to]

zero

one

based

based

k i t k a t

1 →

1 2 3 4 5 6

word[1:3]

-ve →

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

0 → 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Internationalization
1 → 1 2 3 h 5 6 + 8 9 10 11 12 13 14 15 16 17 18 19 20

international 0:13

nation 5:11

intera 0:5

national 5:13

intern 0:6

ion 8:11, 17:
on 18:20, a:11, 18:

String Interpolation in Python

when we write a string without

breaking for including the
variables

a = 10

b = 20

message = "Value of a is " + a + " and value of b is " + b;

msg = f" Value of a is {a} and value of b is {b};"

Indentation

{ } X

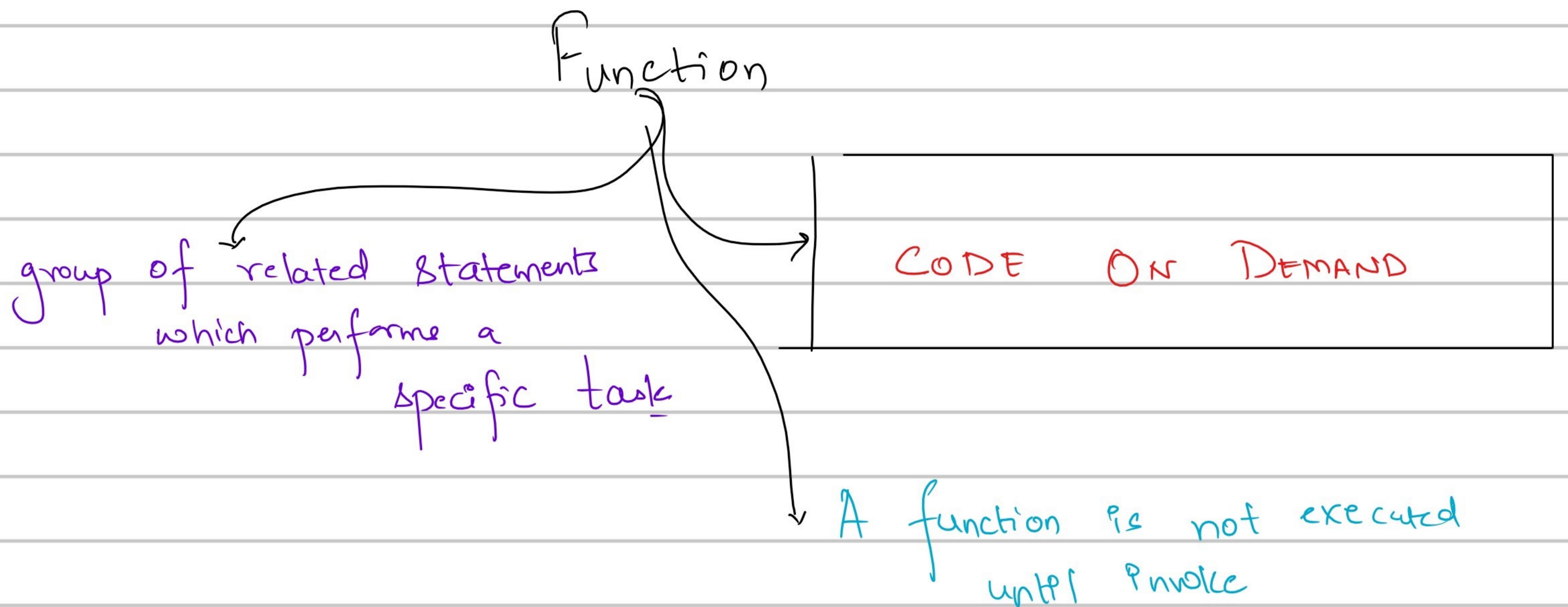
if condition:

 statements

 → space

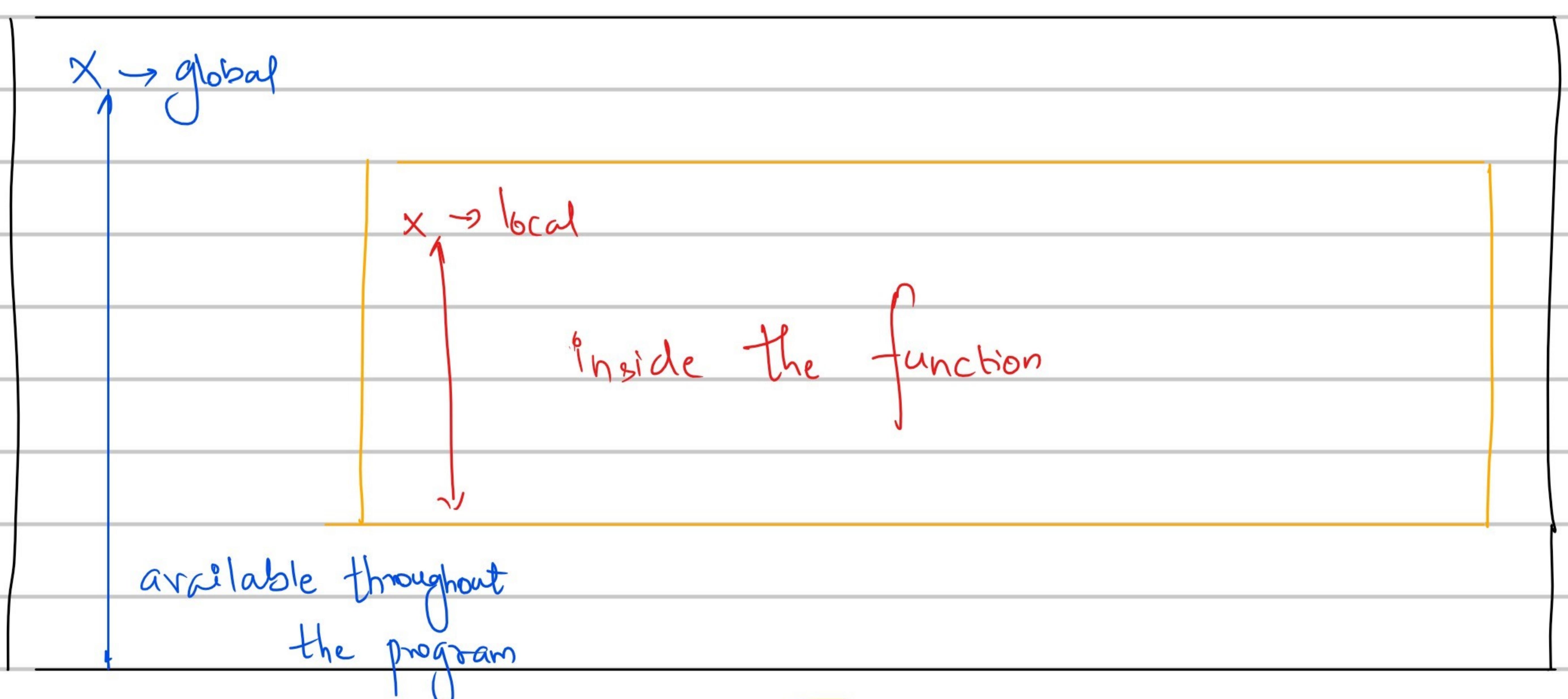
else:

 statements



```
def function_name(parameters):
    """ doc_string """
    statements
    return SomeValue; | optional
```

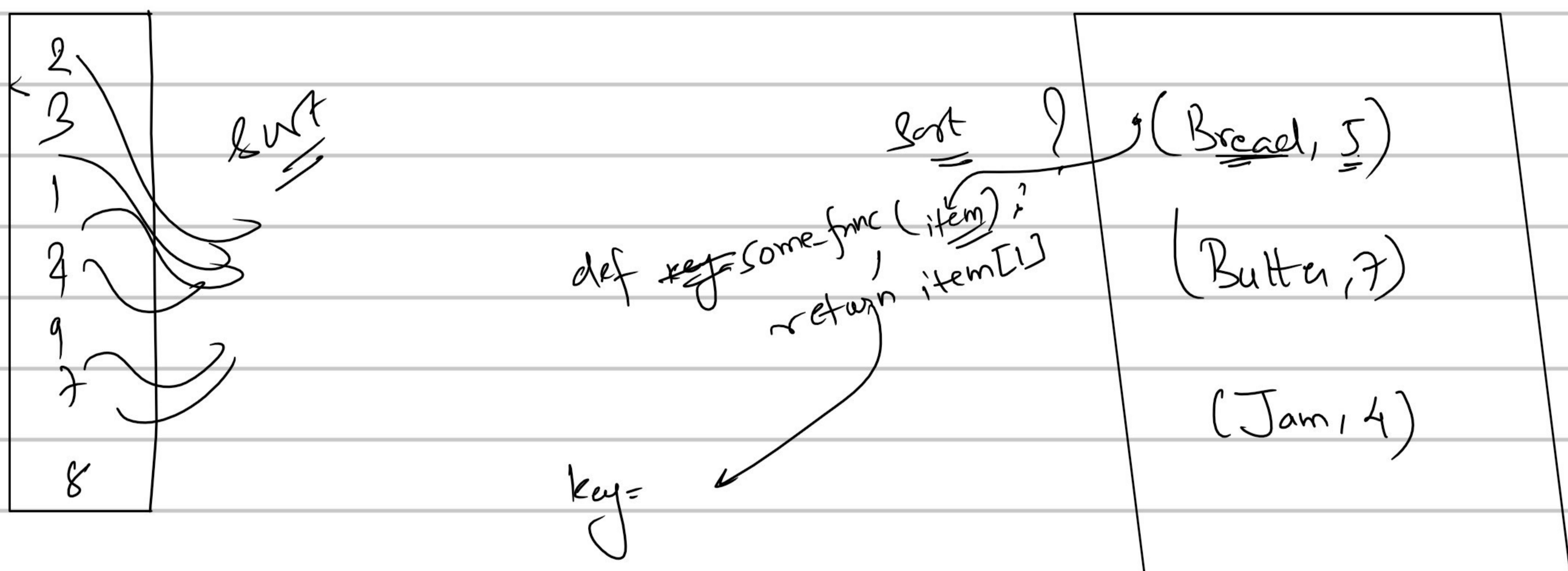
Scope in Python



Python doesn't support block scope; Min scope it supports is function

Data Structures

list tuple range set dict
↓ ↓ ↓ ↓ ↓
[] () range() { } { key : value }



Lambda Functions

↳ anonymous functions

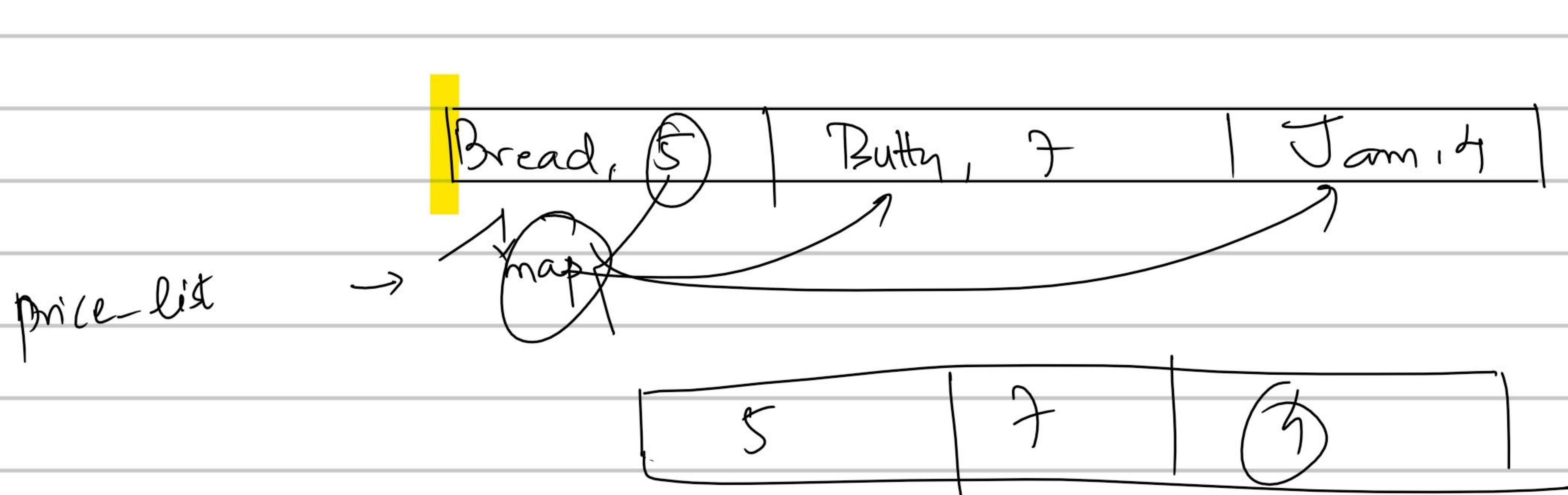
Structure of Lambda Function

lambda parameters : expression

→ no need of return keyword

↓
it executes or
if returns

A lambda function can be assigned to a variable
or it can be passed on to a function



Exceptions

try:

any piece of code
which is prone to
throw exception

except :

unexpected events that
breaks the flow
of code

* Modules *



helps to segregate the files

--main--

python module| package

def



- ① Create a folder
- ② Add __init__.py file

* Regular Expression *

Regex

Zartab@codewithz.com ✓

zartab@abc@xyz.com ✗

zartabOn@gmail.com ✗

→ finding | matching | replacing

patterns in string.

Regex

Basic

Extended

Basic Set

Symbol	What it represents?
*	Zero or more occurrences of the character that precedes this asterisk
.	A wild card that represents any character <small>(alpha-numeric - & special characters)</small>
\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

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

$*$ → 0 or more occurrences

foaaaaabar

fooabar

foobar

fooaabbar

fooxxxbar

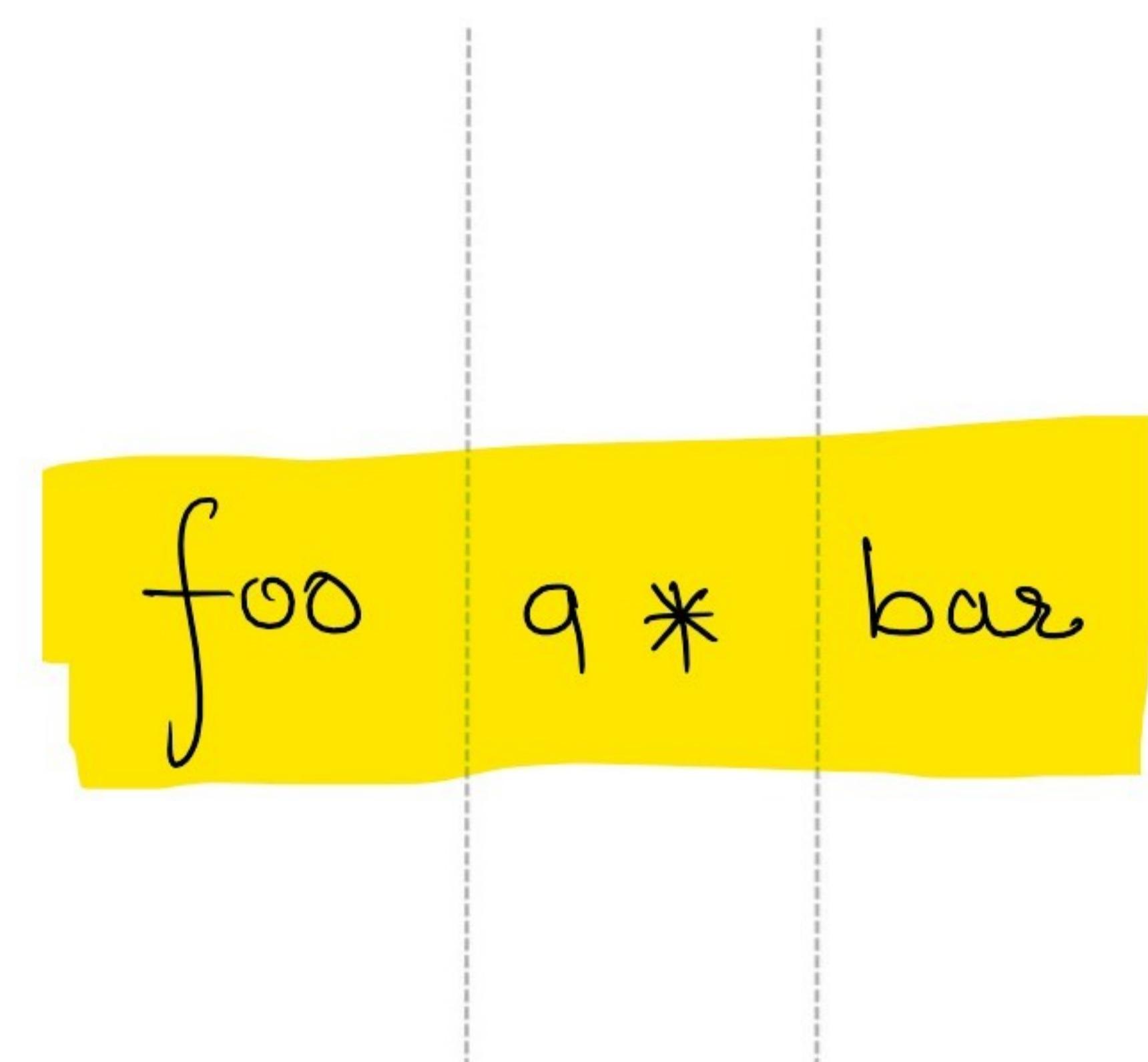
fooxbar

foo aaaa bar

foo a bar

foo bar

foo aa bar



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

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

\cdot → any character

fooabar

fooxbar

baryfoo

foobar

fooxybar

foocbar

foo (a) bar
 foo (x) bar
 foo (c) bar

The diagram illustrates the decomposition of the regular expression `foo . bar`. It shows the string `foo . bar` divided into three segments by vertical dashed lines. The first segment is `foo`, the second is `.` (which is highlighted with a yellow box), and the third is `bar`.

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

1

Understand the Requirement: What needs to be included or excluded

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

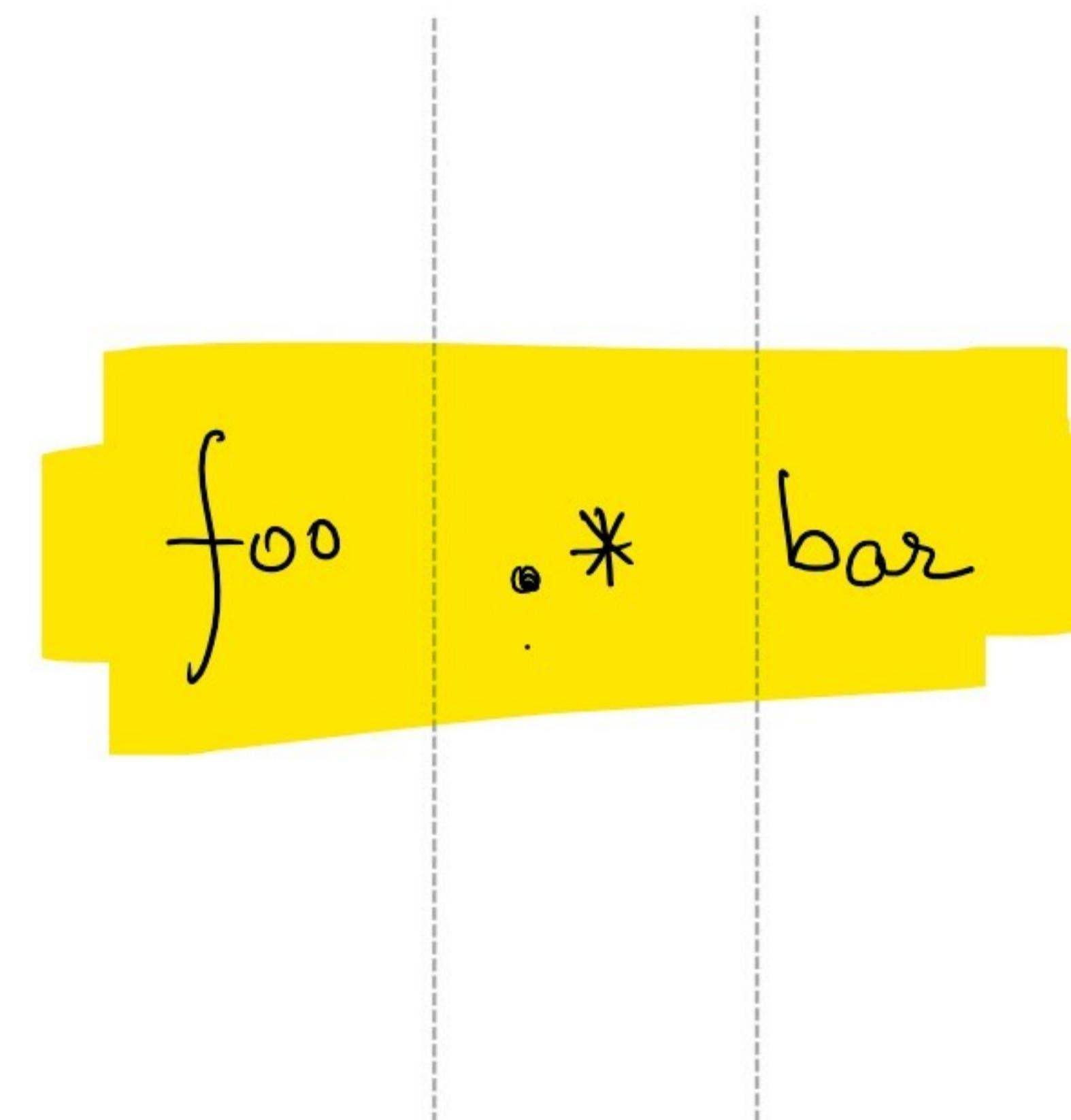
2

Identify the pattern in inclusion or exclusion list

<input checked="" type="checkbox"/> foo		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

- fooxxxbar
- foo bar
- fooxbar
- fooxxbar
- foo bar
- foo bar
- foobar
- fooyyybar

2

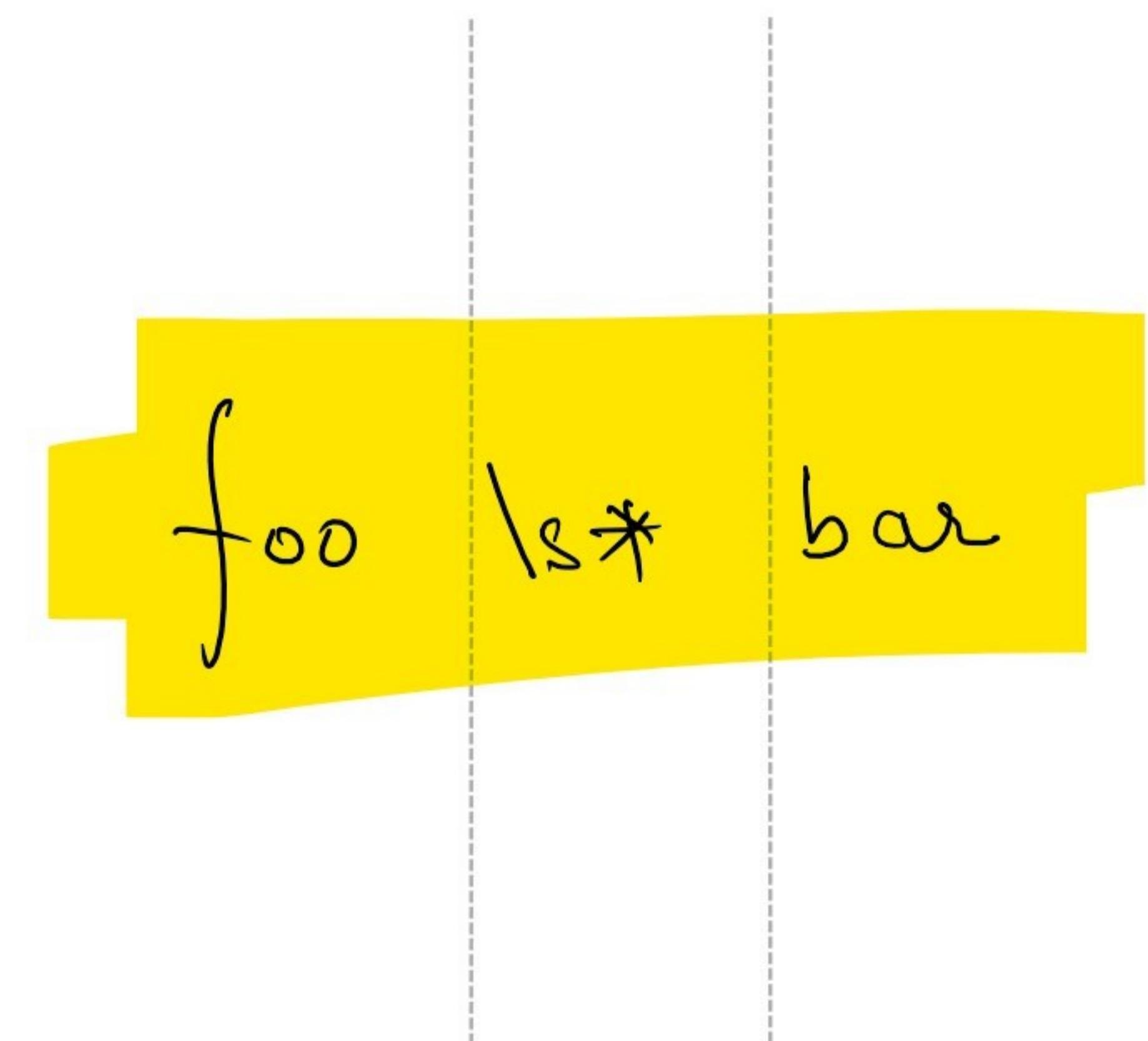
Identify the pattern in inclusion or exclusion list

<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

3

Final Regular Expression

\s-space



\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

$[abc]$ → either a, b, or c

- foo
- moo
- coo
- doo
- poo
- loo
- boo
- hoo

f oo
 c oo
 l oo

$[fc]oo$

$[abc]$ - 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

$[\^ab]$ → should not have a or b

- foo
- moo
- coo
- doo
- poo
- loo
- boo
- hoo

m oo
 h oo

$[fc\^plb]oo$

$[\^mh]oo$

$[\^abc]$ - 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

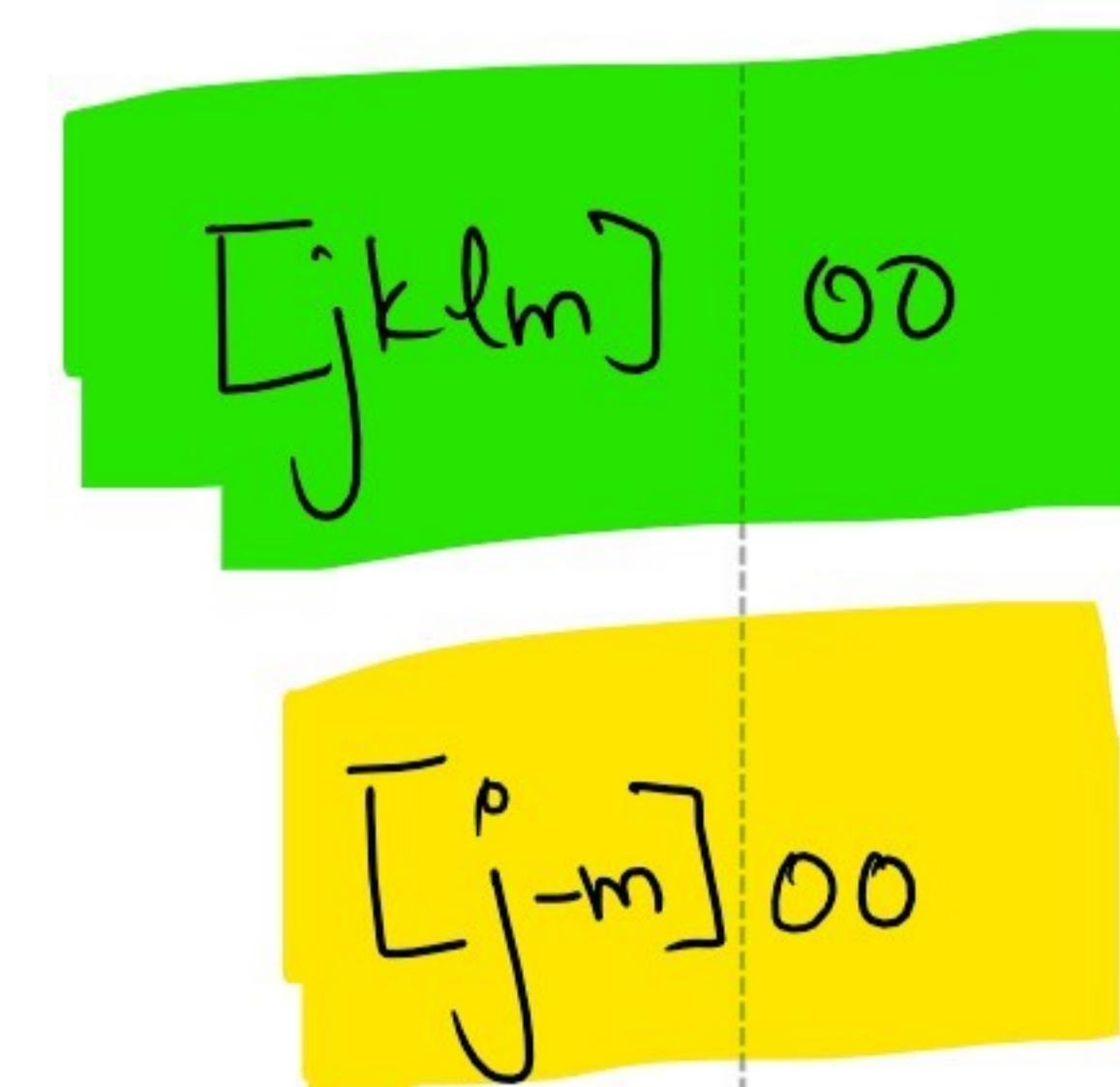
3

Final Regular Expression

$[j-m]$ → anything in j, k, l, m

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



[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

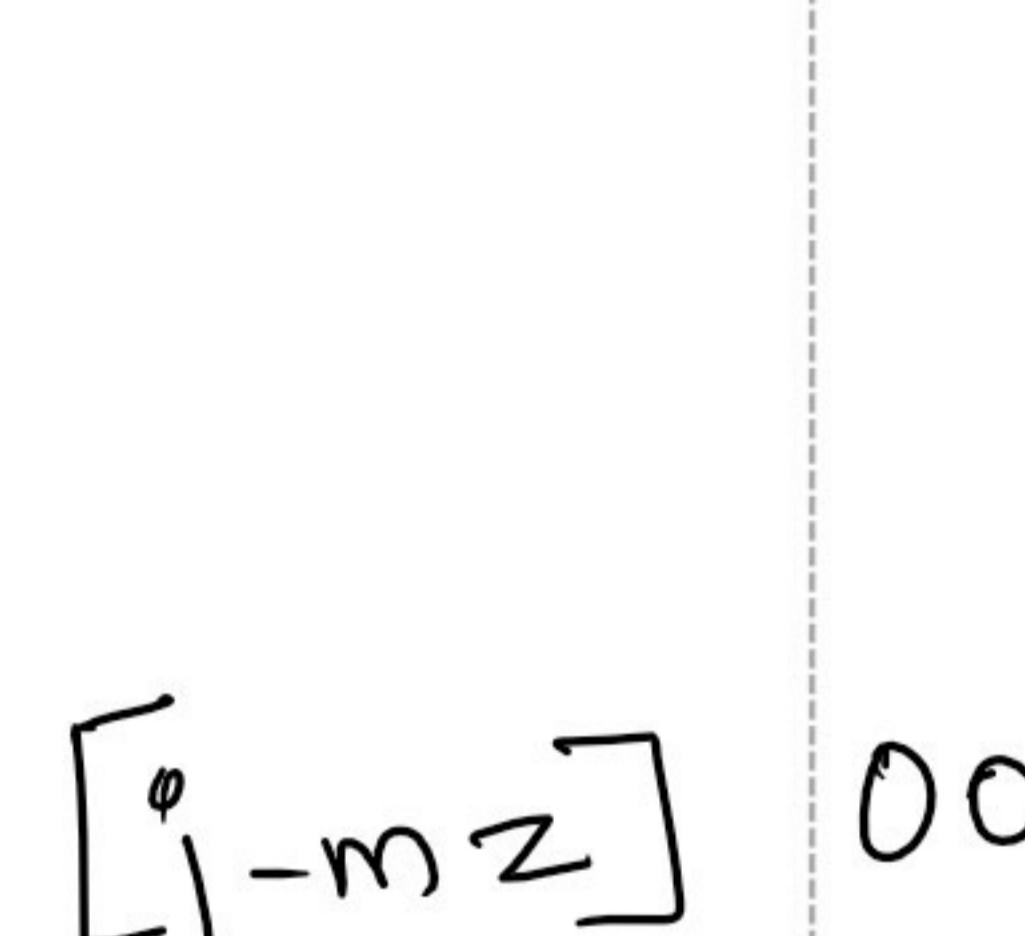
3

Final Regular Expression

$[j-mz]$ → $[j, k, l, m, z]$

- 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



[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

$[j-k-l-m-J-K-L-M-z] \rightarrow j, k, l, m, J, K, L, M, z$

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

- j oo
- K oo
- L oo
- m oo
- z oo

$[j-k-l-m-J-K-L-M-z]$ oo

[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

2

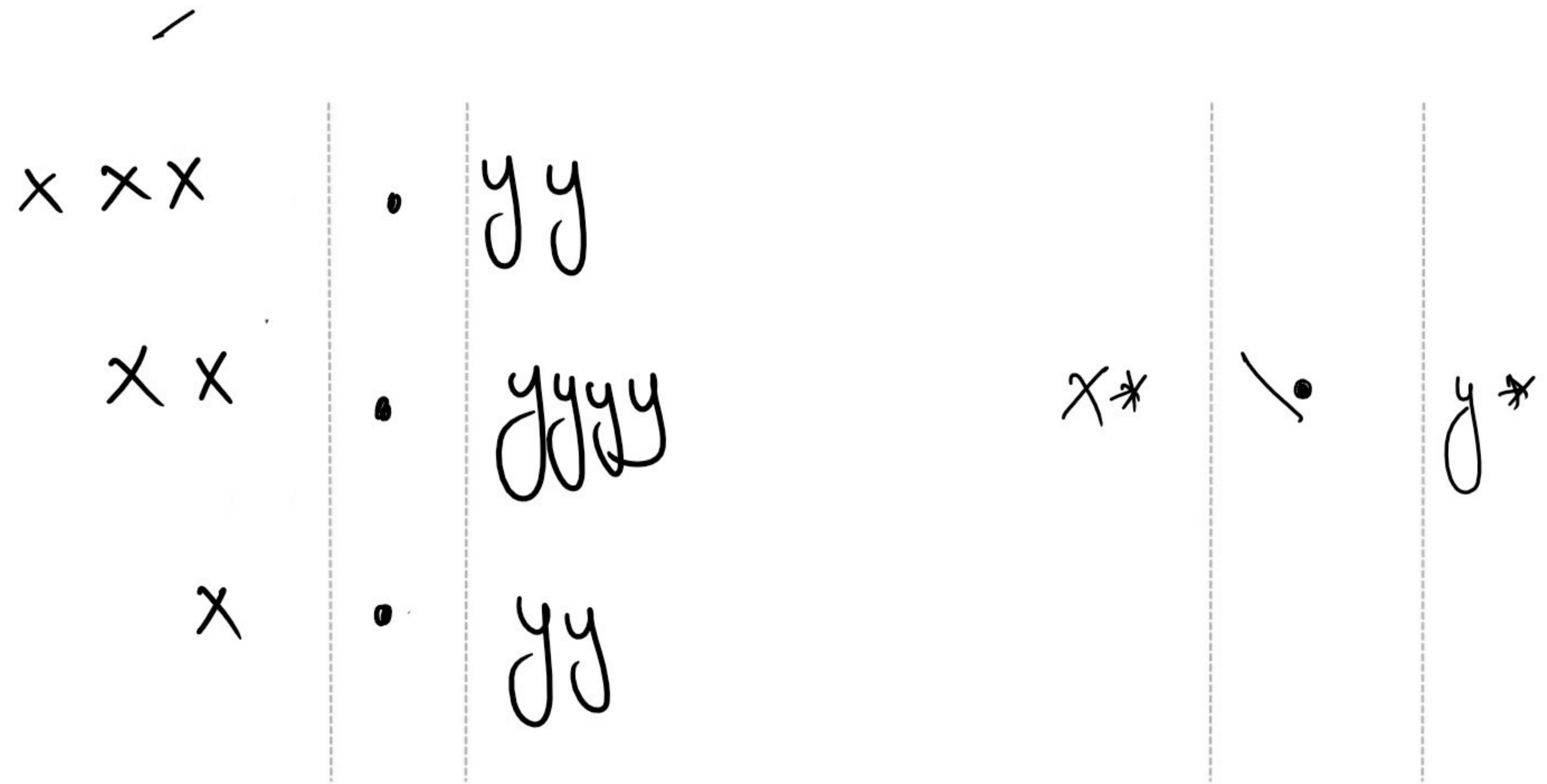
Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

\o → use this as an actual period

- xxx.yy
- xx.yyyy
- x.yy
- xy
- xx.yy
- yy xx
- y x
- yxxx



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

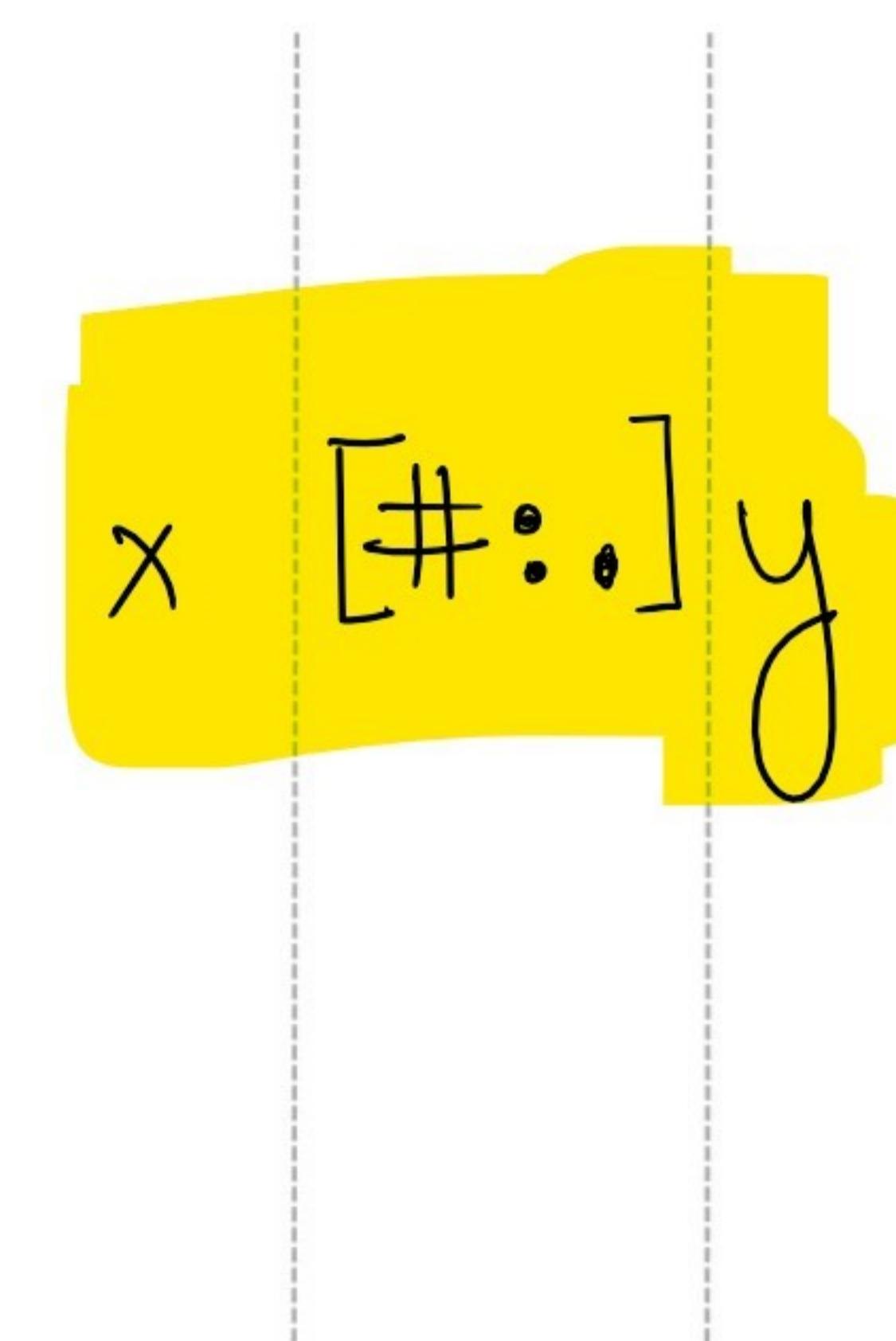
x&y

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

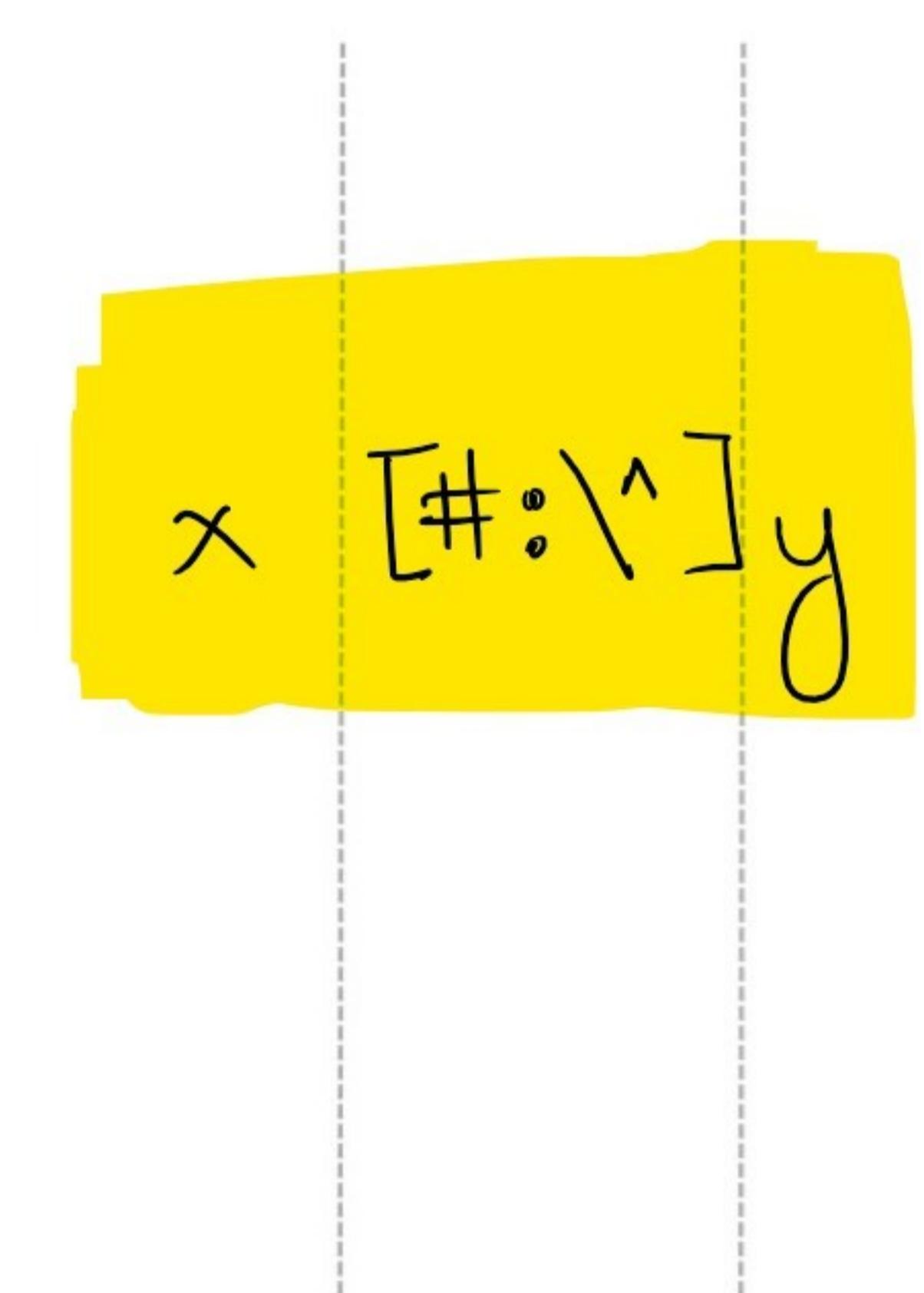
x&y

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

x [# \| ^] y

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

foo bar baz

foo bar baz

bar foo baz

foo baz bar

baz foo bar

foo baz bar

baz bar foo

^ f oo * !

^ is a placeholder that signifies beginning of line. Inside [] a ^ means negation, but outside brackets it acts as an anchor

foo bar baz
 bar foo baz
 baz foo bar
 bar baz foo
 foo baz bar
 baz bar foo

baz foo bar

foo baz bar

→ bars\$

\$ is a placeholder which signifies end of line

Extended Set

Symbol	What it represents?
+ - n	One or more occurrences of the character that precedes + symbol
? 0 - 1	Zero or one occurrence of the character that precedes + symbol
pat1 pat2	Match either the pattern pat1 or pattern pat2
()	Divides the pattern into groups
{m}	Exactly 'm' occurrences of whatever precedes
{m,n}	At least m and at most n occurrences of whatever precedes

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

- 834
- 519
- 4874
- 5
- 89
- 45687
- 25
- 645

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	[0-9][0-9][0-9]	\$
	<input checked="" type="checkbox"/>	[0-9][0-9][0-9]	\$
	<input checked="" type="checkbox"/>	[0-9][0-9][0-9]	\$

$^ [0-9]\{3\} \$$

a{m} represents exactly 'm' repetitions of whatever immediately precedes this

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

- lion
- tiger
- leopard
- fox
- kangaroo
- cat
- mouse
- cuckoo
- deer

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	[a-z]{4}	\$
	<input checked="" type="checkbox"/>	[a-z]{5}	\$
	<input checked="" type="checkbox"/>	[a-z]{5}	\$
	<input checked="" type="checkbox"/>	[a-z]{6}	\$

$^ [a-z]\{4,6\} \$$

a{m,n} represents atleast 'm' and atmost 'n' repetitions of whatever immediately precedes this

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

- ha
- hahahahaha
- hahaha
- hahahaha
- haha
- hahahahaha
- hahahahahahaha
- hahahahahahaha

- ha{5}
- ha{4}
- ha{6}
- ha{8}
- ha{9}

$(ha)^{[4,9]}$

parenthesis is used for grouping and treating as a single entity
 $\{m,n\}$ represents atleast m repetitions of whatever precedes

1

Understand the Requirement: What needs to be included or excluded

2

Identify the pattern in inclusion or exclusion list

3

Final Regular Expression

$+ - |$ or more repetition

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

<input checked="" type="checkbox"/> foo	aaaa	bar
<input checked="" type="checkbox"/> foo	a	bar
<input checked="" type="checkbox"/> foo	aa	bar

foo at bar

a+ One or more occurrences of 'a'

1

Understand the Requirement: What needs to be included or excluded

- https://website
- http://website
- httpss://website
- httpx://website
- httpxx://website

2

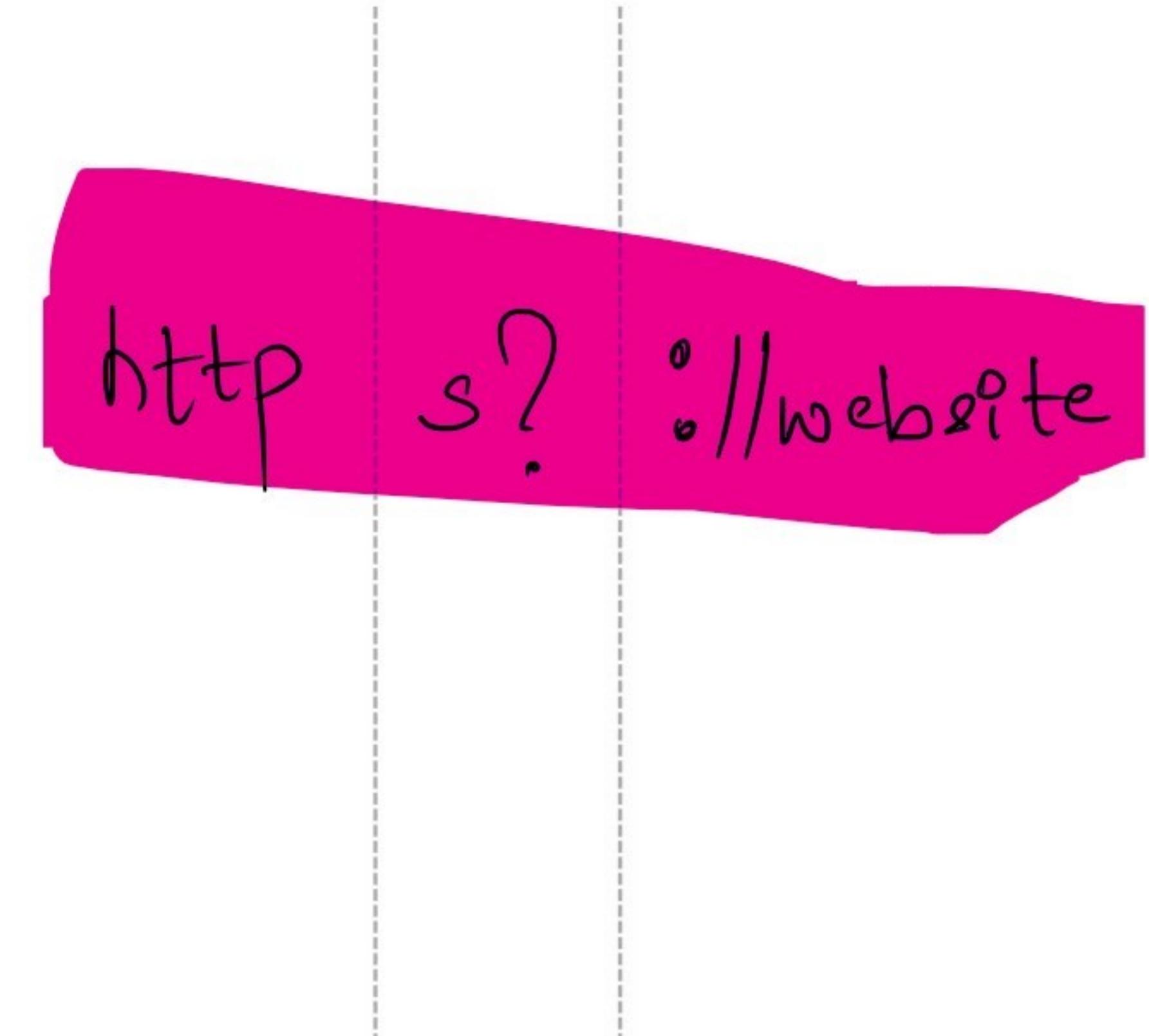
Identify the pattern in inclusion or exclusion list

<input checked="" type="checkbox"/> http	s	://website
<input checked="" type="checkbox"/> http		//website

3

Final Regular Expression

? - 0 - 1



a? Zero or One occurrence of 'a'

1

Understand the Requirement: What needs to be included or excluded

- redwood
- logwood
- sapwood
- rosewood
- plywood
- teakwood

2

Identify the pattern in inclusion or exclusion list

<input checked="" type="checkbox"/> log	wood
<input checked="" type="checkbox"/> ply	wood

3

Final Regular Expression

(ply|log) wood

(a|b) represents either a or b, where a and b can be multi character strings

\A - Matches if the specified characters are at the start of a string.

Expression	String	Matched?
\Athe	the sun	Match
\Athe	In the sun	No match

\Z - Matches if the specified characters are at the end of a string.

Expression	String	Matched?
I like Python	I like Python	1 match
Python\Z	I like Python Programming	No match
Python\Z	Python is fun.	No match

\b - Matches if the specified characters are at the beginning or end of a word.

Expression	String	Matched?
	football	Match
\bfoo	a football	Match
\bfoo	afootball	No match
	the foo	Match
foo\b	the afoo test	Match

\B - Opposite of \b. Matches if the specified characters are **not** at the beginning or end of a word.

Expression	String	Matched?
	football	No match
\Bfoo	a football	No match
\Bfoo	afootball	Match
	the foo	No match
foo\B	the afoo test	No match
foo\B	the afootest	Match

\d - Matches any decimal digit. Equivalent to [0-9].

Expression	String	Matched?
\d	12abc3	3 matches (at 12abc3)
\d	Python	No match

\D - Matches any non-decimal digit. Equivalent to [^0-9].

Expression	String	Matched?
\D	1ab34"50	3 matches (at 1ab34"50)
\D	1345	No match

\s - Matches where a string contains any whitespace character. Equivalent to [\t\n\r\f\v].

Expression	String	Matched?
\s	Python RegEx	1 match
\s	PythonRegEx	No match

\S - Matches where a string contains any non-whitespace character. Equivalent to [^\t\n\r\f\v].

Expression	String	Matched?
\S	a b	2 matches (at a b)
\S		No match

\w - Matches any alphanumeric character (digits and alphabets). Equivalent to [a-zA-Z0-9_]. By the way, underscore _ is also considered an alphanumeric character.

Expression	String	Matched?
\w	12&" : ;c	3 matches (at 12&" : ;c)
\w	%"> !	No match

\W - Matches any non-alphanumeric character. Equivalent to [^a-zA-Z0-9_].

Expression	String	Matched?
\W	1a2%c	1 match (at 1a2%c)
\W	Python	No match

Email Pattern

1. It should have one @ symbol in between
2. Before @ it can be any number of a-z,A-Z,_,.
3. After @ there should be minimum one number of a-z,A-Z,0-9,-
4. After @ and previous combination there should be minimum one period(.) sign
5. It should end with either com,org or net

Jartab.n@gmail.com

Indian Mobile Number Pattern

1. It should have 10 numbers
2. Starting number should be either 7,8, or 9

$^{\wedge} [A-Za-z] [a-zA-Z0-9_+] + @ [A-Za-z][A-Za-z0-9_-+] + \backslash . (com|org|net|edu)$

Email Pattern

1. It should have one @ symbol in between
2. Before @ it can be any number of a-z,A-Z,_,.
3. After @ there should be minimum one number of a-z,A-Z,0-9,-
4. After @ and previous combination there should be minimum one period(.) sign
5. It should end with either com,org or net

Indian Mobile Number Pattern

1. It should have 10 numbers
2. Starting number should be either 7,8, or 9

$[7-9]\{1\} [0-9]\{9\}$

* Object Orientation *

```
numbers = [1, 2, 3]  
print(type(numbers))
```

```
numbers.append(4); ✓
```

```
numbers.fly(); ✗
```

< class 'list' >

blueprint of what all things
a variable which belongs to
a ~~for~~ class of list can do.

fly() is not defined inside the
class

point

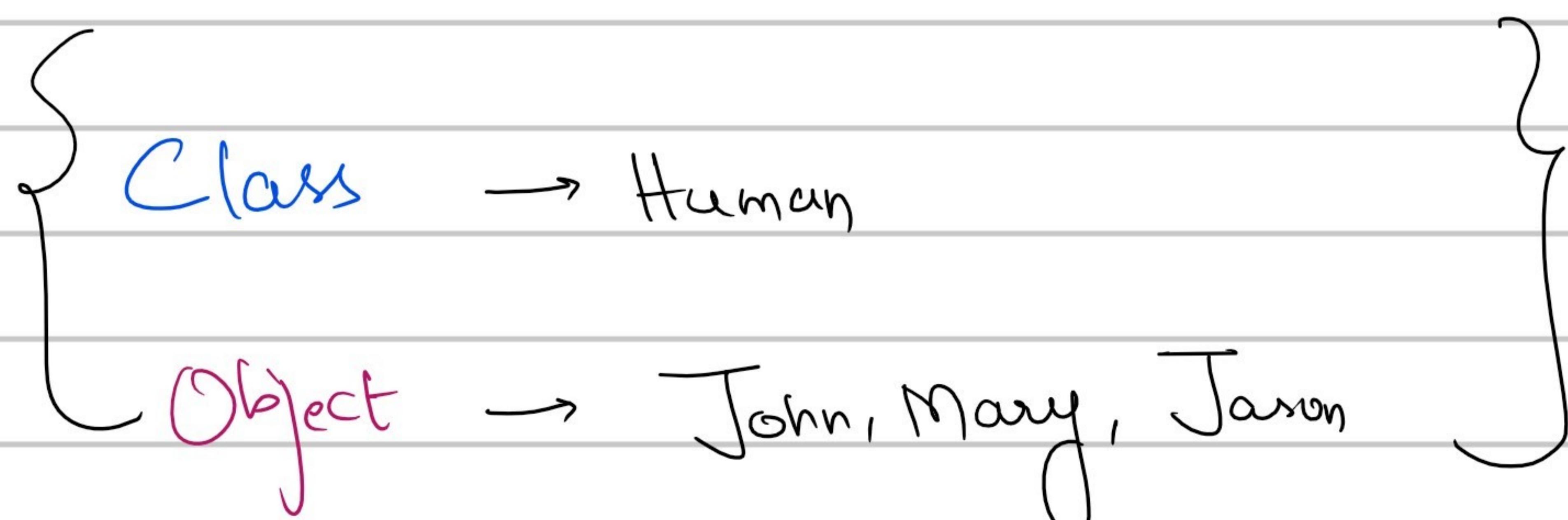
- draw()
- move()
- get_distance()

shopping-cart

- add()
- remove()
- get_price(product)

Class → template or a blueprint which defines the features an object will hold

Object → instance of a class



class Point:

```
def draw(self):  
    print("Something");
```

↑ reference to the object
of the class.
variables will be
assigned to self.

self is not a keyword

→ It is compulsory to pass
a self referring object

* Constructor *

class Point:

```
def __init__(self):
```

representation of
constructor in Python

