Python Functions-Methods: String & List + Operations

Introduction to Computer Programming (Python)

Week 2

Vivatsathorn Thitasirivit

Rev. 1.0 (Course 1/2023)

https://vtneil.com

Recaps

- Simple Process
- Language Levels
- print() function
- input() function
- Escape Characters
- Arithmetic Operators
- Variables
- Data Types
- Type Casting
- Program Flow

Review: Mathematics

Functions

Usually, when we say function, it means something that takes inputs and spits an output.

For instance, a simple quadratic function:

$$f(x) = x^2$$

takes an input x and returns x^2 as an output.
For example,

$$f(2) = 4$$
$$f(3) = 9$$
$$f(-2) = 4$$

This shows that we can take any numbers as an input because squaring is multiplying with itself, which is defined under basic number operations.

As we learn higher mathematics, a function is called a "map" because it maps something to other things.

You will see this definition of a mapping:

$$f : \mathbb{R} \to \mathbb{R} : f(x) = x^2,$$

which means a map f sends real number to real number, defined by $f(x) = x^2$.

This notation is widely used in algebra fields, constructing a foundation to more complex concepts of maps and morphisms.

Review: Mathematics

Functions

A function may take many inputs and spit an output, such as,

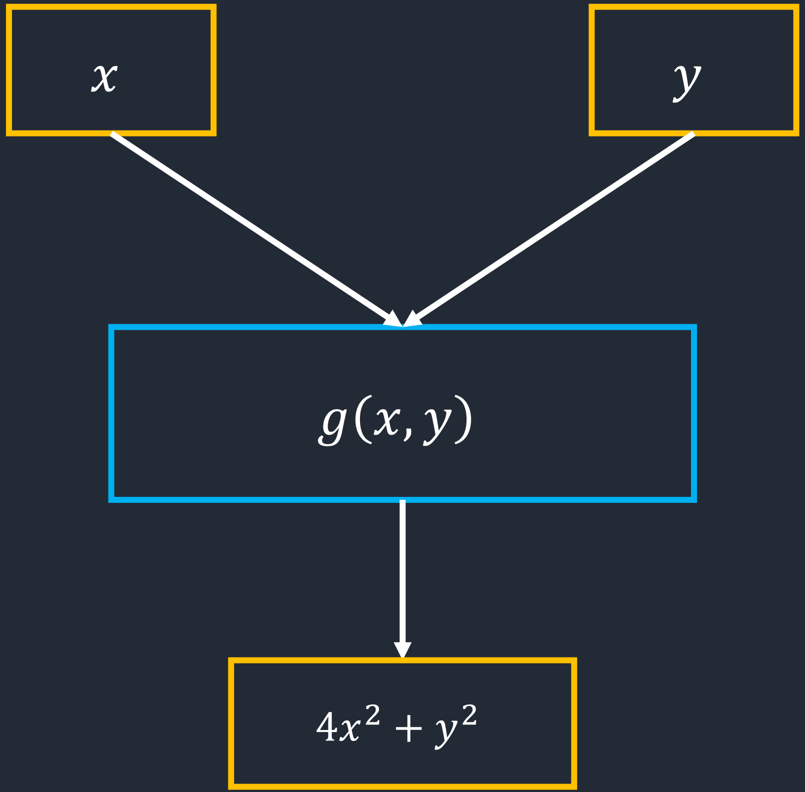
$$g(x, y) = 4x^2 + y^2.$$

Map g is a function as well.

It can also be written as

$$g : \mathbb{R}^2 \to \mathbb{R} : g(x, y) = 4x^2 + y^2$$

because it sends 2 real numbers to 1 real numbers.



Python

Functions

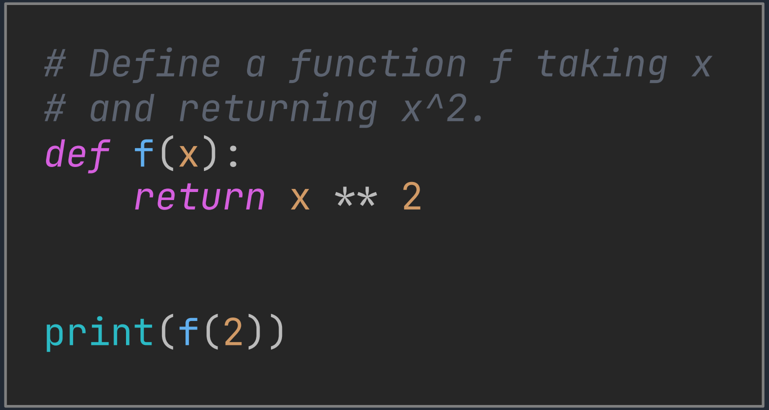
In Python, a function has the same concept as in mathematics but with many more possibilities, such as number of outputs, ***data types***, etc.

A function takes some number of inputs, process them in some way a programmer defined, then output them in the same or different way.

We define a function in Python with “def” keyword, follows with name, inputs (parameters) template, and a colon.

Inside, we will have a return statement as an output.

Sometimes, inputs are called “parameters,” or even “arguments.”



```python
# Define a function f taking x
# and returning x^2.
def f(x):
    return x ** 2


print(f(2))
```

Python

Functions Equivalence Statement

Behind the scene, what computer understands when we called a function with our parameters as follows.

```python
# This:
def f(x, y, z):
    return (x + y + z) / 3


print(f(1, 2, 3))
```

equiv.

```python
# is equivalent to:
x = 1
y = 2
z = 3

print((x + y + z) / 3)
```

Python: Exercise

Functions

Exercise 1.1

Task:

1. Write a function that takes 2 strings as parameters and output a concatenated string.
2. Assign the output to a variable.
3. Print that variable.

For example,

```
concat('Rick', 'Morty')
```

will output "RickMorty".

Use your knowledge of week 1 and functions.

Exercise 1.2

Task: From Exercise 1.1, try the following.

1. Inputting 2 numbers instead of 2 strings and see what happens.

2. Inputting a string and a number instead of 2 strings and see what happens.

Ask yourself: Why did those happen?

Python: Exercise

Functions

Exercise 1.3

More knowledge: a function does not have to take any parameters and does not have to output anything. Just a plain function that does something.

Task: Write a function that prompt user to input 2 numbers and print product of 2 numbers.

Exercise 1.4

Task:

1. Write a function that prompt user to input 2 numbers and return their difference (after – before).
2. Assign the output to some variable and print it.

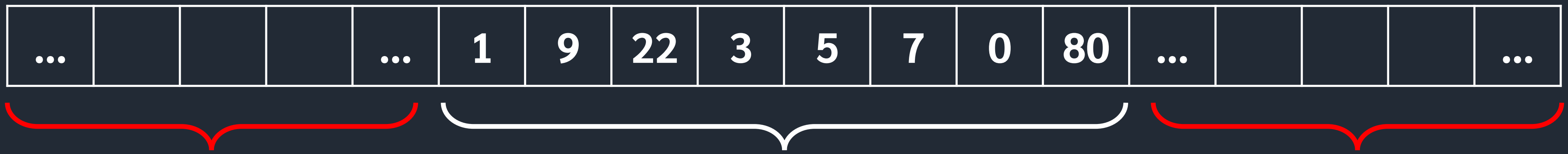
Array

What is an Array? How does it work?

An array is an ordered group of things (any data types).

For example, an array of integers, an array of students,
or even an array of arrays of something.

In computer memory (i.e., RAM), an array is stored in a "contiguous" memory location.



...				...	1	9	22	3	5	7	0	80	...				...

Other memory locations

An array of integers with size of 8.

Other memory locations

Note that the data is stored contiguously.

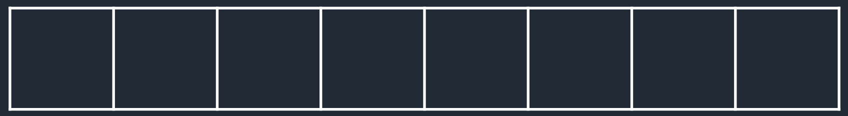
Array

Types of Arrays (by allocation)

There are 2 types of arrays:

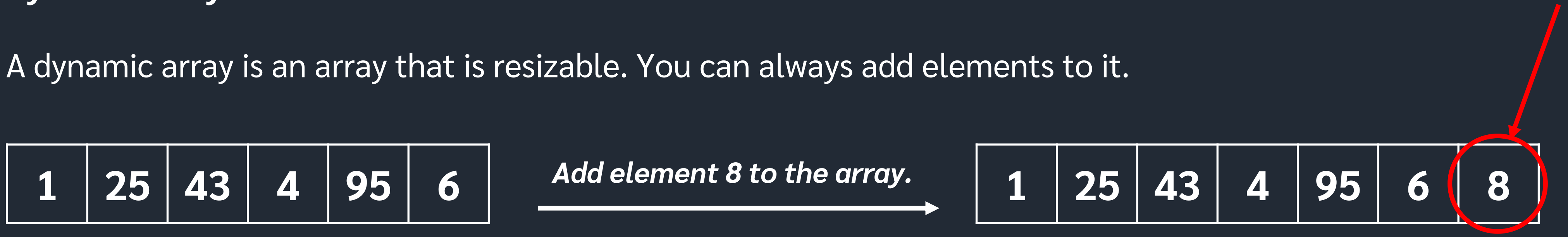
1. **Static Array**

   A static array is an array that is NOT resizable. It must be declared with size value. Once it is created, it has the same size throughout its lifetime.



2. **Dynamic Array**

   A dynamic array is an array that is resizable. You can always add elements to it.



1	25	43	4	95	6

Add element 8 to the array.

1	25	43	4	95	6	8

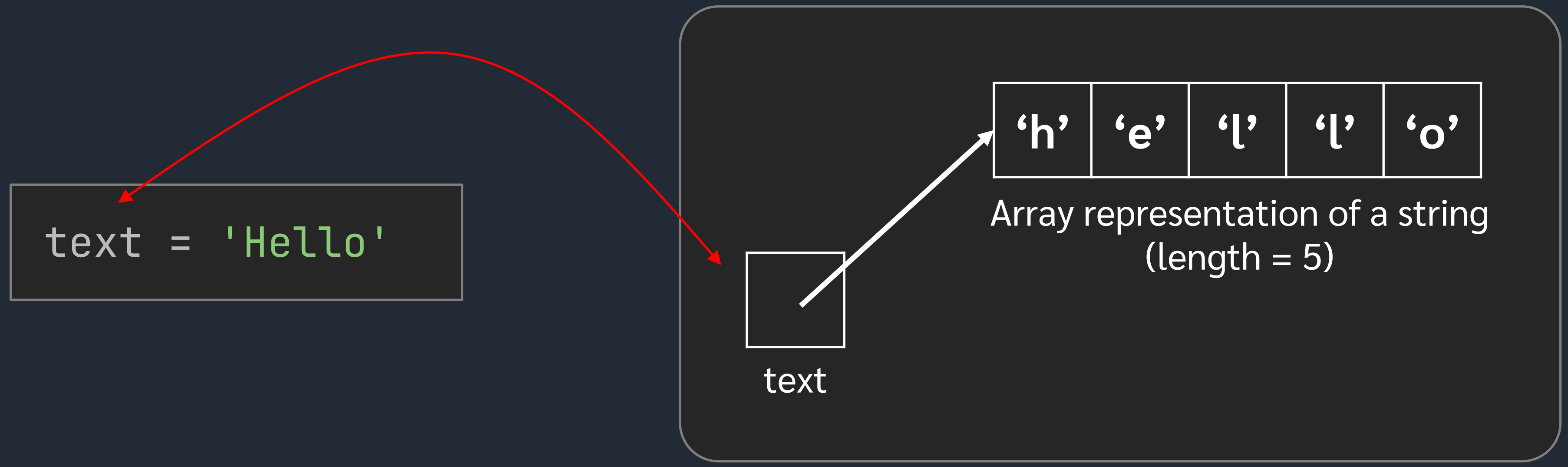
Python

String

A string is just *an array of characters*. String in Python is a “static array.”

Also, string is immutable, *which means that string value can’t be modified.*

For example,



```
text = 'Hello'
```

How the code supposedly works in memory layout

Additional Readings

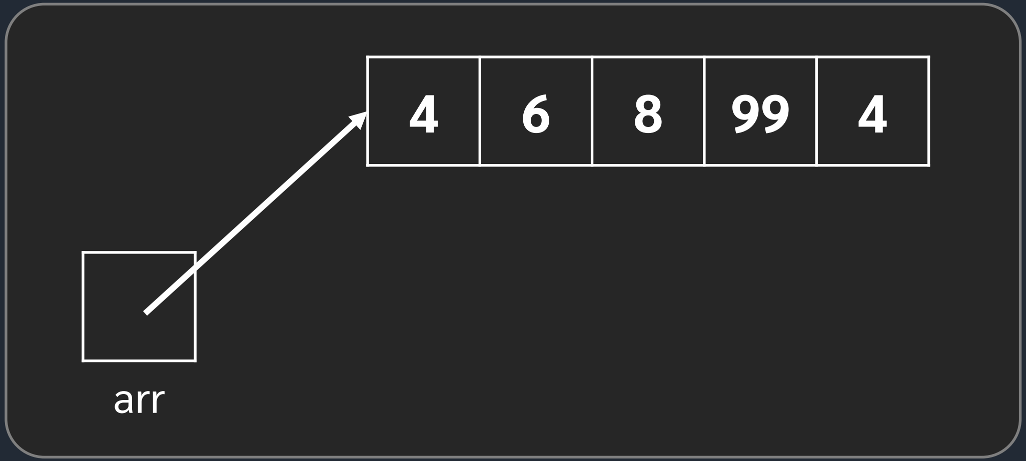
String in other Programming Languages

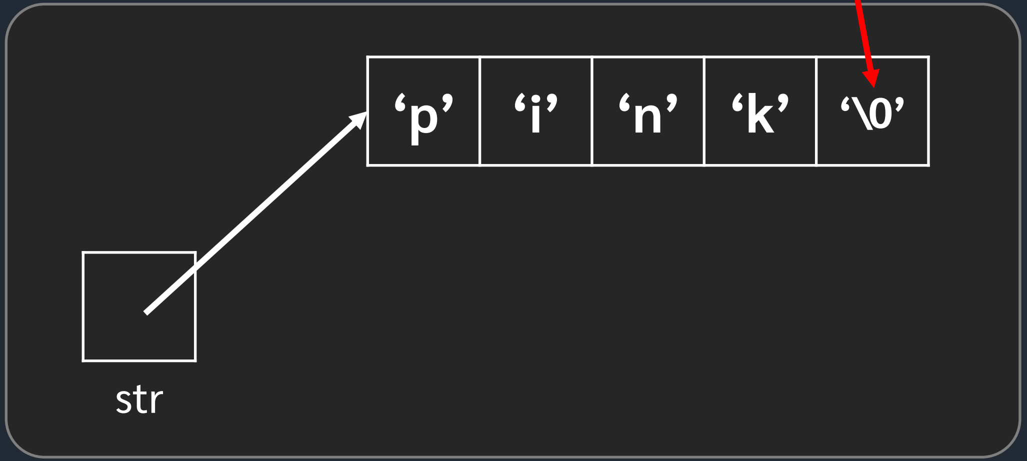
In C, arrays are contiguous block of memory, just like any other languages, but it has very Low-level concept of how you create an array.

These diagram stills represent how an array works in C (more accurate than in Python). Elements in an array in C are stored with the SAME data type.

But!! C does not know the length of the array during runtime. The user must know for themselves what they are doing, or you will be *shooting yourselves in your own foot*.

C uses NULL character to terminate string because C does not know how long the string is, but rather read until finding NULL.





Python: Exercise

Array of Arrays

Exercise 2.1 (Concept)
You learned about array’s structure in computer memory layout format. There can be any depth of nested arrays: an array of arrays of arrays of … .

Draw an array of strings:

1. hello
2. world
3. jkk

(Write the strings in terms of “array of characters.”)

Python

String Indexing & Slicing

In Python, you can get a substring from a string by slicing it.

Recall: Computer starts counting from 0, 1, 2, 3, …, n – 1 with total of n.

Let string s = “Earth”.

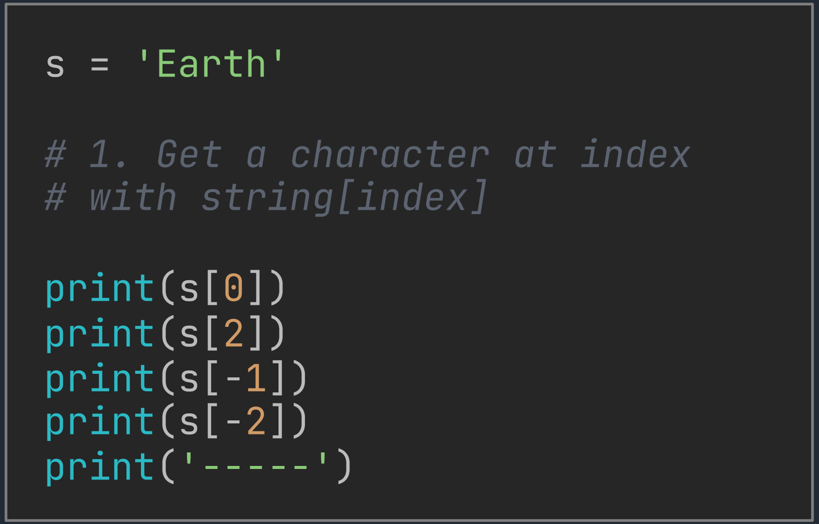
We can slice it in 3 ways with subscript notation [ ].

See next page.

Python

String Indexing & Slicing

1. Get a character at index.

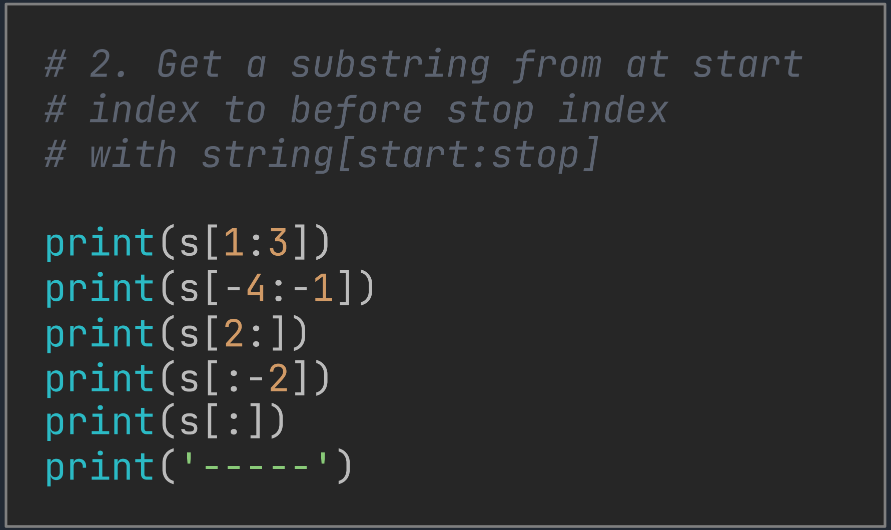


```python
s = 'Earth'

# 1. Get a character at index
# with string[index]

print(s[0])
print(s[2])
print(s[-1])
print(s[-2])
print('-----')
```

Python

String Indexing & Slicing

2. Get a substring from at start index to before stop index. (s is defined last page)

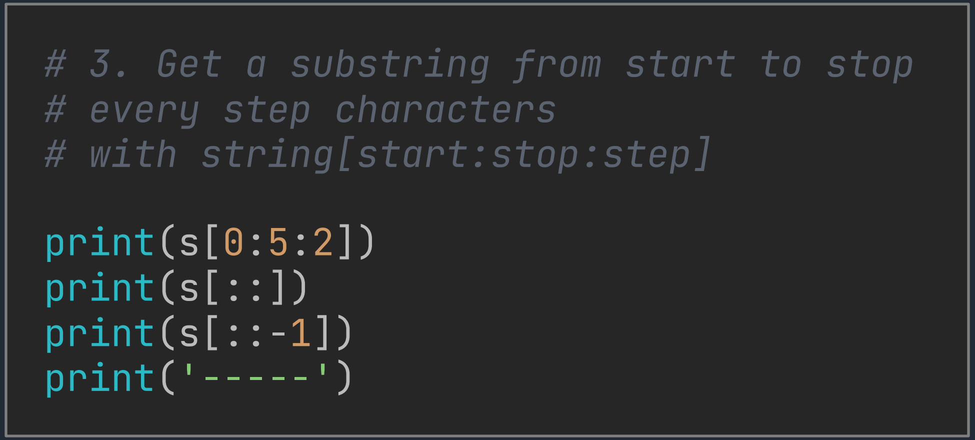


```
# 2. Get a substring from at start
# index to before stop index
# with string[start:stop]

print(s[1:3])
print(s[-4:-1])
print(s[2:])
print(s[:-2])
print(s[:])
print('-----')
```

Python

String Indexing & Slicing

3. Get a substring from start to stop every step characters. (s is defined last page)



```python
# 3. Get a substring from start to stop
# every step characters
# with string[start:stop:step]

print(s[0:5:2])
print(s[::])
print(s[::-1])
print('-----')
```

Python

Methods

In many programming languages, functions and methods are different.

Different in term of usage, but similar in concept.

A method is a function of an object.

What the heck!?

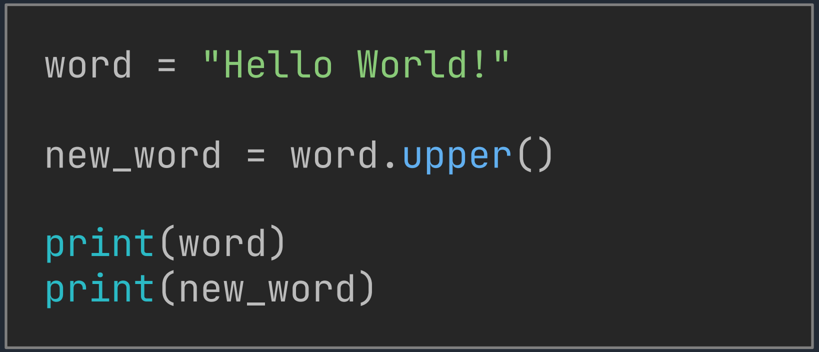
Python

String Methods

In Python, strings are immutable object.

Most of string methods return a new modified string. These are some must-know string methods.

Uppercase everything



```
word = "Hello World!"

new_word = word.upper()

print(word)
print(new_word)
```

Python

String Methods

Lowercase everything

```
word = "Hello World!"

new_word = word.lower()

print(word)
print(new_word)
```

There are many more string methods which you can research them for yourselves, but with limited programming knowledge, I’ll guide you to find some methods.

Python: Exercise

String Methods

Exercise 3.1

In Python, there is a string method that search the whole string
and returns an index of that substring if substring is found.

For example,

“This is just an example sentence”.<method>(“This”) will return 0.

“This is just an example sentence”.<method>(“is”) will return 2.

“This is just an example sentence”.<method>(“ample”) will return 18.

“This is just an example sentence”.<method>(“Joker”) will return -1. (Substring not found)

Task: What is that method? Try googling it!

Python: Exercise

Function & String

Exercise 3.2

Task: Write a function that takes 3 string as an input:

String 0: full word
String 1: old word
String 2: new word

and returns the following:

In case String 0 has String 1 as substrings in it, replace that substrings with String 2.
Then, return the reversed version of that string, all capitalized.

Hint: use string `replace` method to replace substrings.

```python
# Format
def convert(full, str1, str2):
    # Do something
    output = ''
    return output
```

```python
# Example 1
convert('this is the string that is beautiful', 'is', 'are')
# This should output 'LUFITUAEB ERA TAHT GNIRTS EHT ERA SIHT'
```

```python
# Example 2
convert('teststringblablabla', 'b', 'x')
# This should output 'ALXALXALXGNIRTSTSET’
# because it replaces every 'b' with 'x'
```

```python
# Example 3
convert('teststringblablabla', 'txt', 'blablabla')
# This should output 'ALBALBALBGNIRTSTSET’
# because there is no 'txt' in it.
```

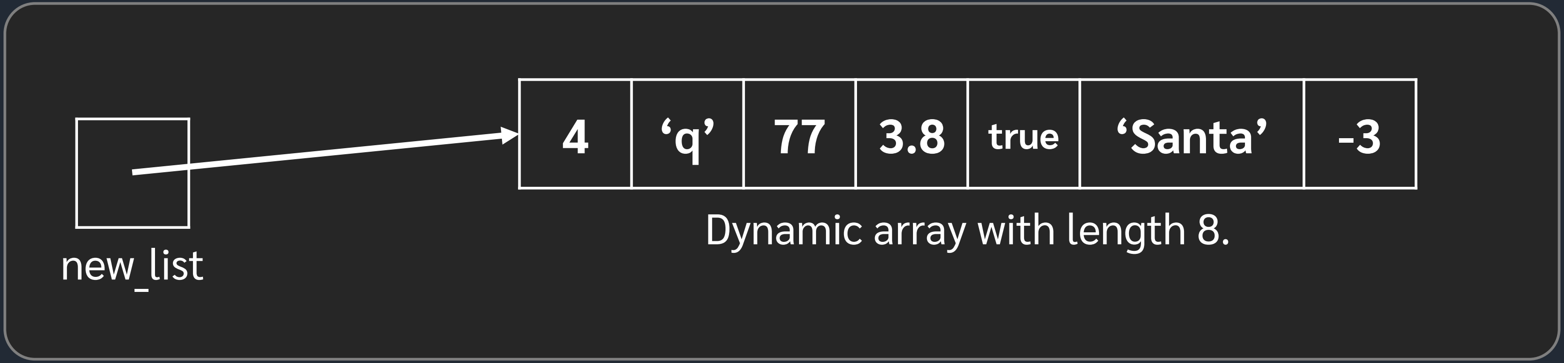
Python

List

A list is *an array of characters*. List in Python is implemented as a “dynamic array,” so it is resizable (Removing and adding elements to list will update its length).

Also, list is mutable, *which means that objects and each object can be modified.*

Elements in list can be different types as list in Python is very versatile.



List memory layout (supposedly)

Python

List Indexing & Slicing

You learned about string indexing & slicing. List indexing & slicing is IDENTICAL.

You can index an element, index a range of elements, index a range with steps, and reverse the list.

Creating a list of elements in Python

```
new_list = [1, 2, 3, 4, 5]

print(new_list)
print(new_list[::-1])
```

Python

List Operations

Like strings, lists can use addition operator to concatenate multiple lists and multiplication with positive integers for list repetition.

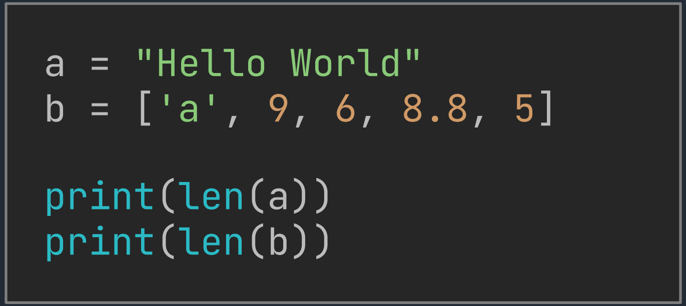
```
a = [1, 2, 3, 4]
b = [4, 4, 3, 3]

c = a + b
d = a * 3

print(c)
print(d)
```

Python

String & List Length

In Python, there is built-in function “len” that returns the length of strings and list.

For example,



```
a = "Hello World"
b = ['a', 9, 6, 8.8, 5]

print(len(a))
print(len(b))
```

Python

List Methods

Lists are mutable, so most of their methods do not return anything, but rather modify their content as specified.

Useful list methods:

1. append(element)	Add an element to the last position of the list.
2. clear()	Clear the list.
3. copy()	Return a copy of that list.
4. count(value)	Count occurrences of that value in the list.
5. extend(another list)	Concatenate list with another list.
6. index(value)	Return an index (position) of that value in the list.
7. insert(position, element)	Insert an element at position to the list.
8. pop()	Remove last element of the list.
9. remove(value)	Remove first occurrence of that value in the list.
10. reverse()	Reverse the list.
11. sort()	Sort the list in ascending order.