Python Beginners workshop

Day 1

About myself

My name is Prakamya Singh, and I'm the Head of Web Development at the GIIS Tech Club.

I have 2 years of experience in programming with Python, and I have been learning Web Development for the past one year. I have additional experience in data science and data analysis with Python.

Contact me(for Python-related queries):

WhatsApp:

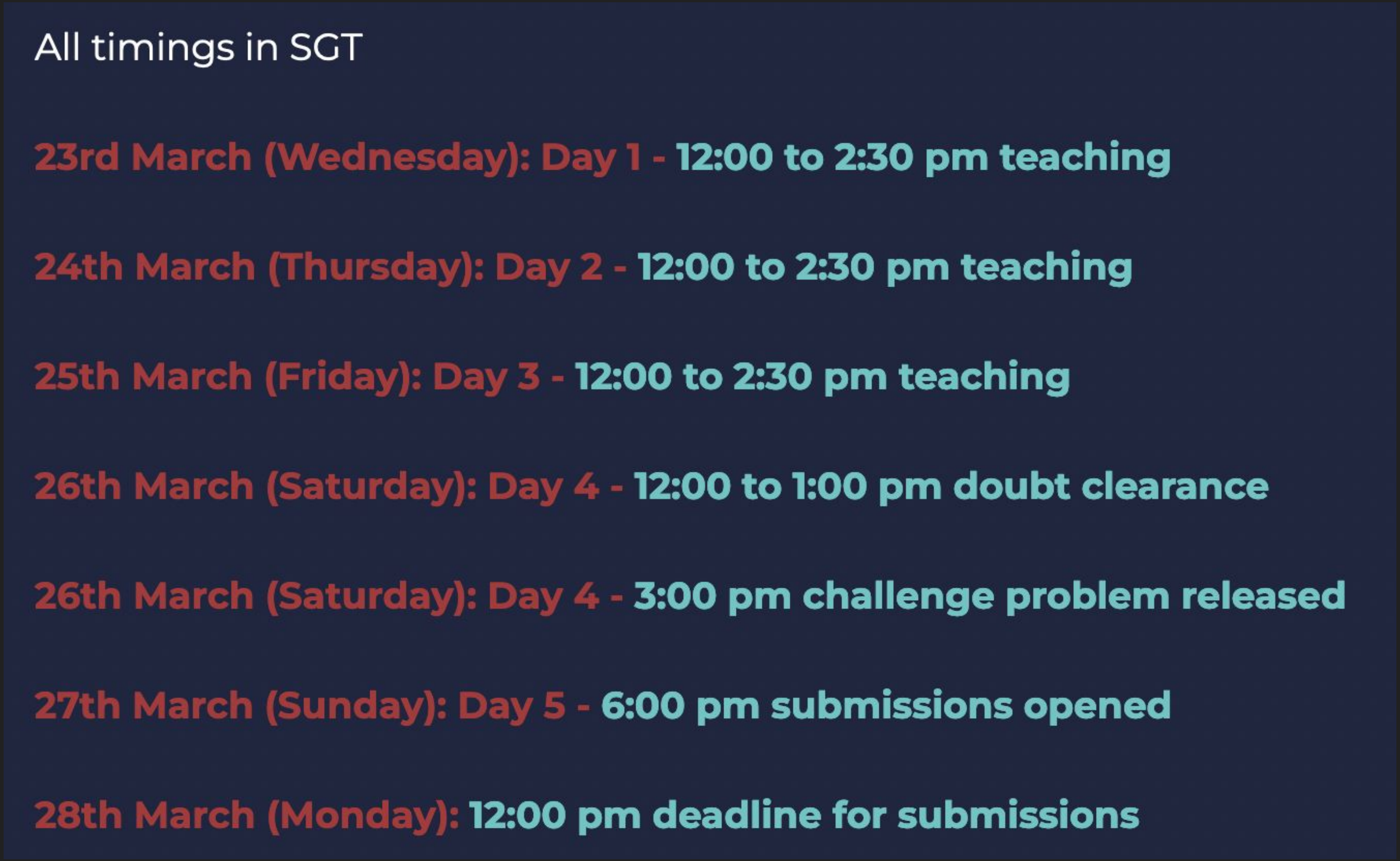
Discord:

About the Workshop

In this workshop, you will learn all the basics of Python - one of the most popular programming languages in the world.

At the end of this workshop, you will be provided with the knowledge to not only solve the challenge problem, but also to create your own programs. This workshop will also provide you with a platform to develop and enhance your programming skills.

Workshop Schedule



All timings in SGT

23rd March (Wednesday): Day 1 - 12:00 to 2:30 pm teaching

24th March (Thursday): Day 2 - 12:00 to 2:30 pm teaching

25th March (Friday): Day 3 - 12:00 to 2:30 pm teaching

26th March (Saturday): Day 4 - 12:00 to 1:00 pm doubt clearance

26th March (Saturday): Day 4 - 3:00 pm challenge problem released

27th March (Sunday): Day 5 - 6:00 pm submissions opened

28th March (Monday): 12:00 pm deadline for submissions

Contact Us

For general queries

Website: https://tech-club.vercel.app/html/contact.html

Instagram: @_giistechclub_

Email: giistechclub2022@gmail.com

WhatsApp:

Workshop code of conduct

- You are free to take screenshots or notes of the slides/code - it is recommended to take notes
- No recording will be shared at the end, but the slides will be shared along with the code snippets made during the workshop
- You can ask questions in the chat or unmute yourself and ask them live
- If you have any doubts after the workshop, you can contact anyone from the Tech Club via any of the contact methods mentioned in the previous slides
- Please do not spam in the chat, and please keep yourself on mute if you don’t want to ask a doubt/talk
- The discord server for the workshop has its own set of rules - please read them and follow them strictly

Cyber ethics

1. No online harassment, racism or hate speech of any kind in any forum - if you see online users doing this, report them to the forum moderator
2. Don’t make any code with malicious intent, and don’t use any code made with such intent - hacking into app servers, spamming in chat forums and overloading networks or servers all count as cyber attacks and may have legal repercussions
3. Be respectful and humble when online - don’t tease people for their lack of knowledge, and don’t show off your knowledge either. Hate speech is even worse
4. Respect privacy and confidentiality
5. Be honest and trustworthy - plagiarism is not good, and could have legal repercussions

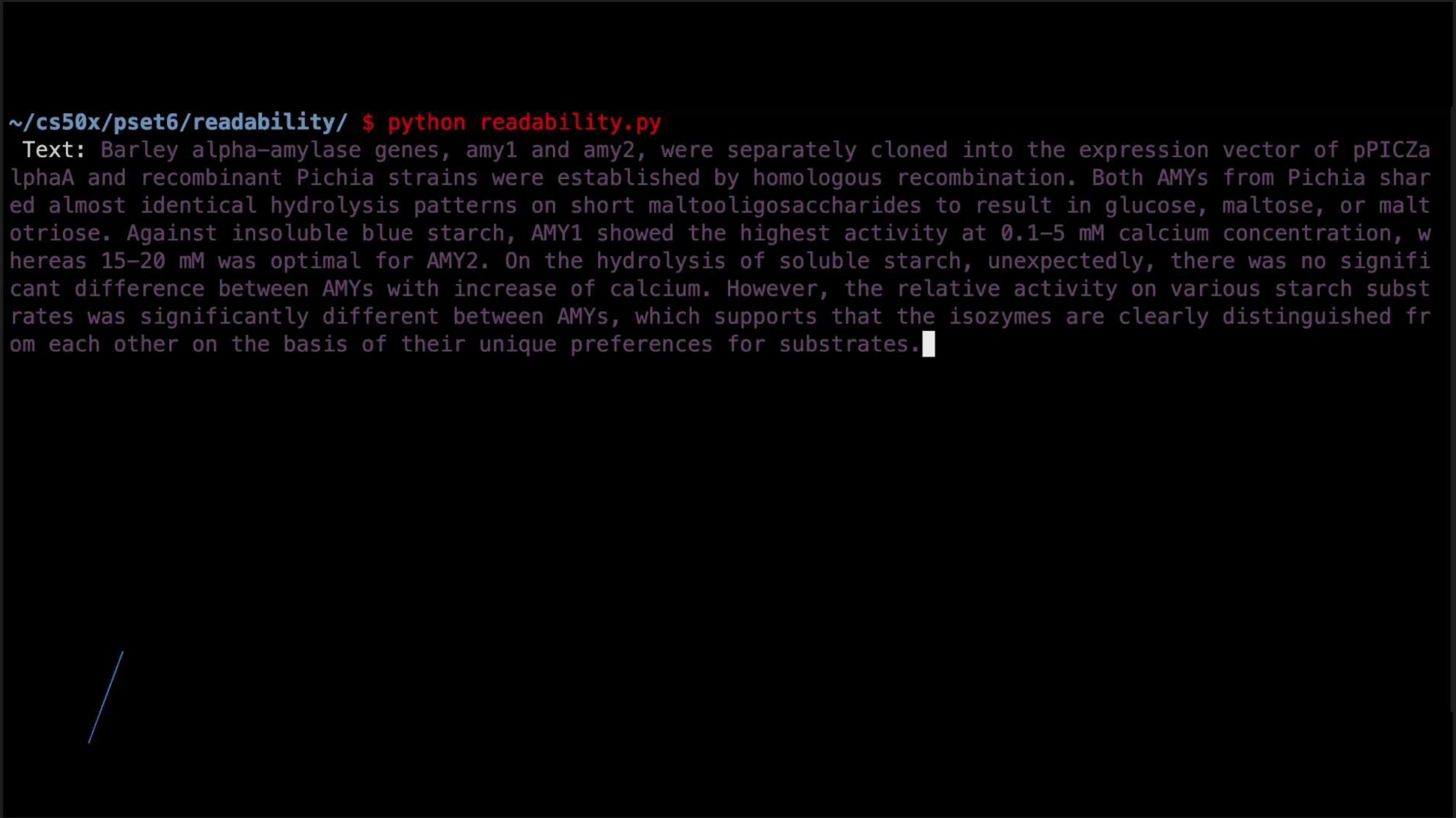
Why Python?

Python is a very simple language to learn and use, because of its english-like syntax.

This however makes it slow to run, so it is not often used in apps or other codes that need to run at high speeds(such as web browsers or video games).

Python is used mostly in data science and data analytics, where fast programs are not essential, and large amounts of data need to be processed in complicated ways.

It is also used in the the field of AI and machine learning.



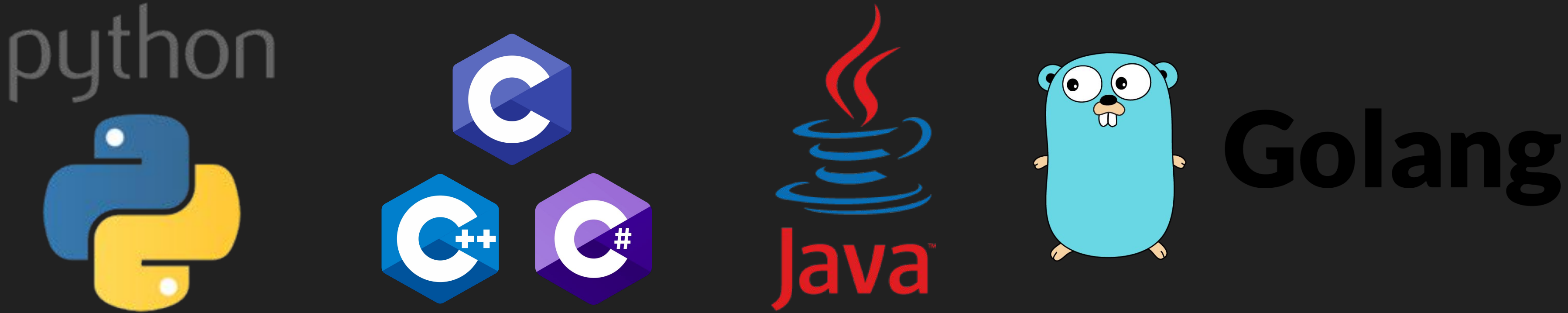
```
~/cs50x/pset6/readability/ $ python readability.py
 Text: Barley alpha-amylase genes, amy1 and amy2, were separately cloned into the expression vector of pPICZa
lphaA and recombinant Pichia strains were established by homologous recombination. Both AMYs from Pichia shar
ed almost identical hydrolysis patterns on short maltooligosaccharides to result in glucose, maltose, or malt
otriose. Against insoluble blue starch, AMY1 showed the highest activity at 0.1-5 mM calcium concentration, w
hereas 15-20 mM was optimal for AMY2. On the hydrolysis of soluble starch, unexpectedly, there was no signifi
cant difference between AMYs with increase of calcium. However, the relative activity on various starch subst
rates was significantly different between AMYs, which supports that the isozymes are clearly distinguished fr
om each other on the basis of their unique preferences for substrates.
```

What is a program?

A program is a **sequence of instructions that automate the performance of a task**.

A programming language is **a medium that allows programmers to write these instructions**. Each programming language has its own syntax, but all programs that perform a particular task will use the same logic(same sequence of steps), no matter what the programming language.

Examples of programming languages include Python, Java, C/C++/C# and Golang



Algorithms

An algorithm is a **sequence of instructions that solves a very specifically defined problem or performs a specific task.**

It is different from a program in that a program performs a general task, whereas an algorithm performs a more specific sub-task that is part of performing the overall task of a program.

For example, take an alarm clock app. The program of the app will be responsible for everything: setting alarms, telling the time, sounding an alarm, setting timers, running stopwatches etc. But an algorithm within the program could be responsible for checking the time, and if the current time matches an alarm time, to sound an alarm.

Functions

Sometimes in a program, some instructions will be repeated many times throughout the program. In these cases, these instructions are grouped together into functions. The function is given a name, and that name is used when that code needs to be run.

So instead of writing a 10 line code repetitively, you can write it once while defining the function, and call it many times throughout the program in just one line.

All programming languages have their own in-built functions, but you can define and create your own functions, and customize them as you like.

Variables

Variables are used in programs to store information. The value of a variable can be changed during the program execution.

You can name a variable anything, but there are certain rules:

- Variables cannot have the same name as functions or keywords of the programming language - a var called while in python will not be allowed
- They cannot have spaces in the name - num 1 is invalid, but num1 is fine
- They cannot use any special characters except for _ - num-2 is invalid, but num_2 is ok
- They cannot start with a number - num3 is fine, but not 3num
- If you want to create a new variable, it cannot have the same name as a variable that you have already made

These naming rules apply to functions as well

Alarm Clock app - variables

```
alarm_hour = …

alarm_minutes = …

alarm_active = …
```

These are some variables that have been created to keep track of alarms in the app. But what values will we assign them?

Basic Data Types

All programming languages assign variable values to specific data types

Based on the assigned data type, a certain group of operations and functions will be allowed on that variable

For example, you can perform arithmetic operations on numeric data types, but not on strings

Integer

This is a numeric data type, and any whole number(negative or positive) is classed as an integer.

Ex: 1, -4, 0, 10

Real number/Float

Both of these are decimal numbers(any numbers with a decimal point). Even whole numbers in decimal form are classed as real numbers.

Ex: -0.12, 4.8, 9.9999…, 9.0

Boolean

These are a special type of numeric data type, that can take only two possible values(usually referred to as True and False). They are very useful in conditions(and loops, conditionals or functions that use conditions)

String

This can be any kind of textual data. Strings are often enclosed in quotes(double or single, does not matter as long as they match), and can contain any unicode characters.

Ex: “hello”, “abc123”, “@#/?|\2-=3;{};”

Array

An array is a collection of items that are all of the same data types

Arrays are usually fixed in length; a new array will have to be made if one or more values need to be added/removed from the existing array

Many programming languages have their own variations on arrays.

Arrays in python are called lists, and are much more flexible in the sense that values can be added or removed easily, and you can put different values with different data types in the same list.

Ex: [1, 2, 4, 3, 5, 7]

Alarm Clock app - assigning values

I could use integers to store the hours and minutes of the alarm:

```
alarm_hour = 6
alarm_minutes = 30
```

I could also have a boolean value to ensure if the alarm is active or not:

```
alarm_active = True
```

I can put all of the above data into an array to reduce the number of variables:

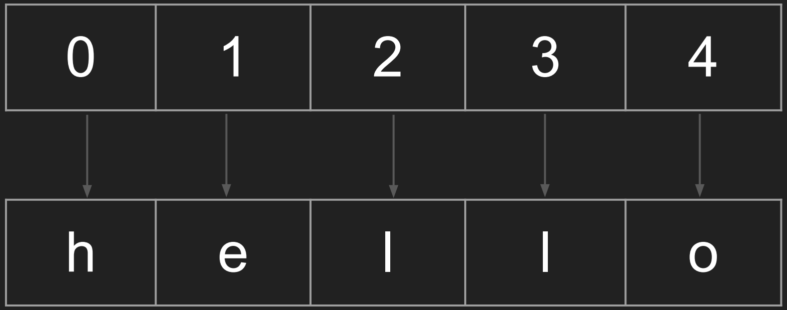
```
alarm_info = [6, 30, True]
alarm_days = [“Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”]
```

Indexing

For strings and arrays, it is sometimes useful and necessary to get individual characters or elements of the string/array.

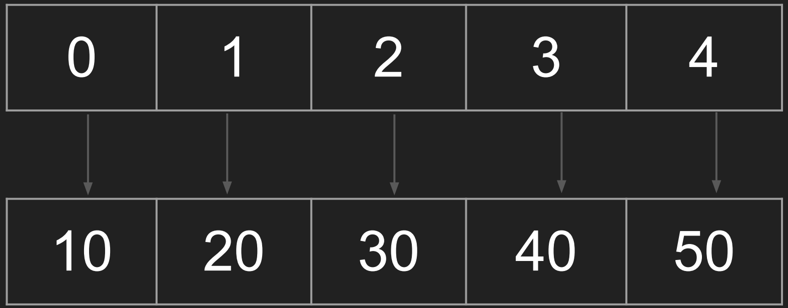
Indexing allows us to do this.

The index of an element is its position in the string/array. In programming languages, indexes start with 0, so the 1st element will have an index of 0, the 2nd element will have an index of 1, and so on.



0	1	2	3	4
h	e	l	l	o

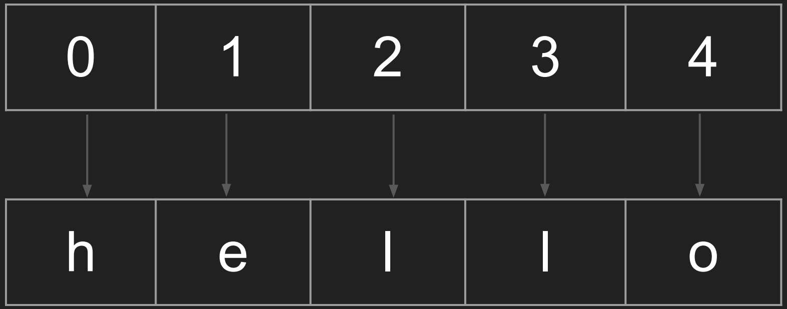
For strings



0	1	2	3	4
10	20	30	40	50

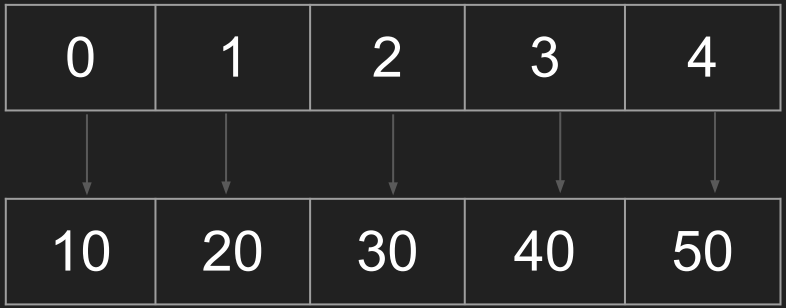
For arrays

If the length of an array is 500, then what will the index of its last element be?



0	1	2	3	4
h	e	l	l	o

For strings

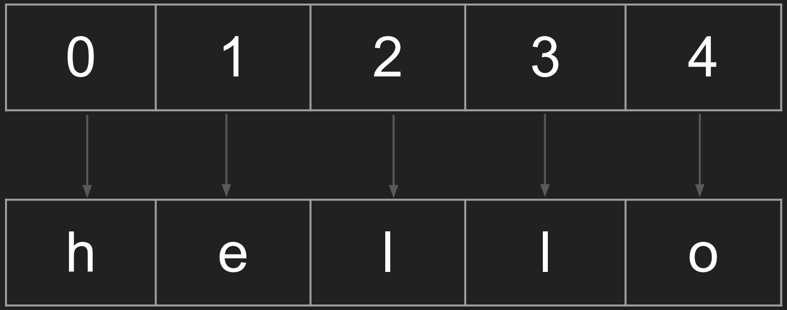


0	1	2	3	4
10	20	30	40	50

For arrays

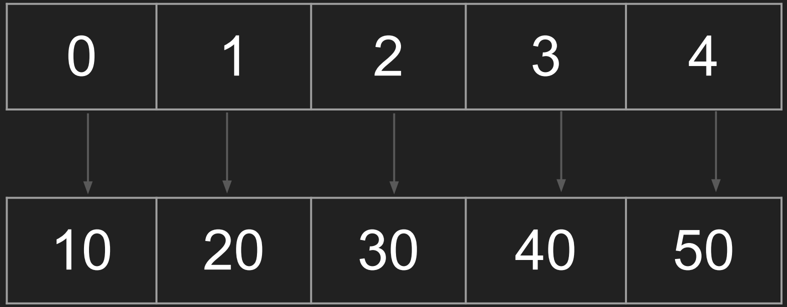
If the length of an array is 500, then what will the index of its last element be?

We can see that if an element is at the 1st position, its index is 0. So, we can deduce that if the position of an element is x, then its index is x - 1. So the last element of an array of length 500 will be at position 500, which means its index will be 500 - 1 = 499



0	1	2	3	4
h	e	l	l	o

For strings



0	1	2	3	4
10	20	30	40	50

For arrays

If the length of an array is L, then what is the index of its last element?

We can see that if an element is at the 1st position, its index is 0. So, we can deduce that if the position of an element is x, then its index is x - 1. So the last element of an array of length L will be at position L, so its index will be L - 1

How to use indexes

If you want to access an element at a certain index, you use []:

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

To get the 3rd position, I need index 2:

```
x[2]
```

This will give 3

You can also access indexes using variables:

```
y = 7
x[y]
```

This will give 8

Input and output

In all programs, you will need to take some data as input, process it using your program, and output the result of the processing.

Input may be in the form of data provided by another file, data transferred over the internet, or data directly from the user.

Output can be in the form of data transferred back over the internet, data entered into a different file, or data printed out and displayed to the user.

Most complex programs use some combination of these input and output forms, but in this workshop we will look at how to use user input and output data only.

Alarm Clock app - Input and output

How do I know what time the user wants the alarm for? I have to take some input from the user:

```
INPUT alarm_time
```

In the background, I will set all the variables…

I may want to tell the user that their alarm was set successfully. I can do this with output:

```
OUTPUT “Alarm set”
```

Conditions

Conditions are statements that will dictate whether a certain thing happens or not.

A block of code that uses conditions are called conditionals.

You can check conditions using IF and ELSE statements in most programming languages. They are used the same way as in normal english sentences:

```
IF something_is_true THEN do something

ELSE do something else
```

Alarm Clock app - Conditions

I will need to check if the current time is equal to the alarm time. If it is, then I will play the alarm sound, otherwise I will ignore.

```
IF current_time = alarm_time THEN
    IF alarm_active = True THEN
        play_alarm_sound
```

Now, when the alarm time is same as the current time, and the alarm is active, then the app will play the alarm sound.

```
IF current_time = alarm_time AND alarm_active = True THEN
        play_alarm_sound
```

Loops

There are two main types of loops: for and while loops

Loops are instructions that will execute repeatedly, until something happens to stop them.

This could be a condition being satisfied, or a counter value being reached, or something external stopping it(user manually closes it, or device runs out of battery for example).

While loops

A while loop will run endlessly as long as a given condition is satisfied.

It can be written in this format:

```
WHILE something is true
    do something
```

While the condition is true, the code inside the loop will be working, but once the condition is dissatisfied, the loop will end and the rest of the program will continue

Be careful not to start an infinite loop with a condition that will never be dissatisfied:

```
WHILE 3 > 2
    do something
```

Since 3 is always going to be > 2, the loop will never end

Alarm Clock app - while loops

Once the alarm rings, it needs to keep ringing on repeat until the user turns it off.

```
WHILE user_hasnt_turned_off_alarm
    play_alarm_sound
```

You may want to account for situations where the user has lost their phone somewhere, and you might want to switch off the alarm automatically after 15 min:

```
WHILE user_hasnt_turned_off_alarm AND current_time - alarm_time < 15
    play_alarm_sound
stop_alarm_sound
```

For loops

In a for loop, a counter variable whose value changes with each repetition of the loop is needed. The counter variable goes from a starting value to an ending value

```
FOR counter = 1 TO 100
      OUTPUT counter
```

You can customize the counter variable to skip a certain number of values

```
FOR counter = 2 TO 100 STEP 2
      OUTPUT counter
```

Comments

In a program, you may want to write some short lines of explanation of the code that will explain what each part is doing(in case someone else is going to look at it, or you come back to your code after a long time and forget). To prevent the computer from treating these lines as regular code, we use comments.

Ex:

```
# Takes input from user

INPUT number
```

# is the symbol used to define a commented line in python, but some other programming languages may use // (C, C++, C#, Java, JS) or something else

Importance of syntax

Syntax is basically the “grammar” of a program. Computers can only look at the code and read the exact lines you have written, and cannot guess what you were trying to write if you made a mistake.

Any spelling mistakes or capitalisation errors will lead to the program giving a Syntax Error.

```
while # actual keyword, will work fine

whlie # wrong spelling, will give error

While # wrong capitalisation, will give error
```

Syntax errors causes

1. Wrong capitalization or spelling(as shown before) of keywords or variables
2. Forgetting to add a comment symbol:

   ```
   Takes input
   INPUT number
   ```

3. Not closing brackets or quotes properly:

   ```
   “Hello    # quotation mark not closed
   “Hello’    # starts with double-quotes but ends with single-quotes
   x = 4 * (y+10))    # extra bracket
   ```

4. Missing punctuation or missing keywords:

   ```
   x = 4y  # missing an * for multiplication
   ```

Logical Error

A logical error will allow the program to run, but the output of the program will not be expected. A simple example is as follows:

```
# I want to input a number, add 2 to it and output the result
INPUT num
OUTPUT num * 2
# If I input 4, I will get 8 as output, but I wanted 6
```

To fix it, I have to change the output statement: OUTPUT num + 2

Basic terminal commands

First, open up the terminal or command prompt on your computer.

To see what files and folders are in the current directory, use the ls command(for macOS) or the dir command(for Windows).

To change your terminal directory location, use the cd command: cd <folder name>

To go back to the home directory, run cd again: cd ~

To create a new folder within the current directory, use the mkdir command:

```
mkdir <folder name>
```

Installing Python

First, download Python from https://www.python.org/

Make sure you have downloaded the right files based on your OS (Windows or MacOS)

Then, open the terminal/command prompt on your computer.

Then type in python3 -V (for MacOS) or py -V (for Windows). You will see the version of python(Python 3.10.2) if it has been installed successfully.

Installing pip - Package Installer for Python

First, check if pip was already installed by typing pip -V (same for Windows and MacOS)

The latest version of pip is 22.0.4; make sure this is the version you are using(or at least 21.0.0 and above). If you get a message asking you to get the newer version of pip, run this command: pip install --upgrade pip

If you don’t have pip installed, you will receive some sort of error message saying “command not found: pip”. In this case, try using pip3 instead of pip

If that doesn’t work, then install this file and run python3 get-pip.py (for MacOS) or py get-pip.py (for Windows) after cd-ing into the downloads folder(or wherever the file was installed. Once pip has been installed, you can delete this file.

Installing and Setting Up Visual Studio Code

First, download the latest version of VS Code from https://code.visualstudio.com/

Make sure you have downloaded the right version based on your OS(MacOS or Windows)

Open up VS Code(after following the set-up instructions) and install the Python extension(the one by microsoft, you will see a blue check mark next to it) by searching it up on the extensions marketplace.

You can go online and look for different extensions or themes to install or use, but only the Python extension will be enough for this workshop.

When using python, you will have to choose the python interpreter(VS Code will ask you to choose). Choose the version that you have just installed (Python 3.10.2)

Resources

Link to downloading Python

Installing Python on MacOS

Installing Python on Windows

Install get-pip.py

Install VS Code

Setting up Python extension

Setting up Python for VS Code