

Python Exercises Report

Hussein Hussein
Student ID: 58301

Repository Information

GitHub Repository: <https://github.com/Leo-devv/cloud-programming-labs>
Online Editor: <https://replit.com/@replit-userhy/cloud-programming-labs>

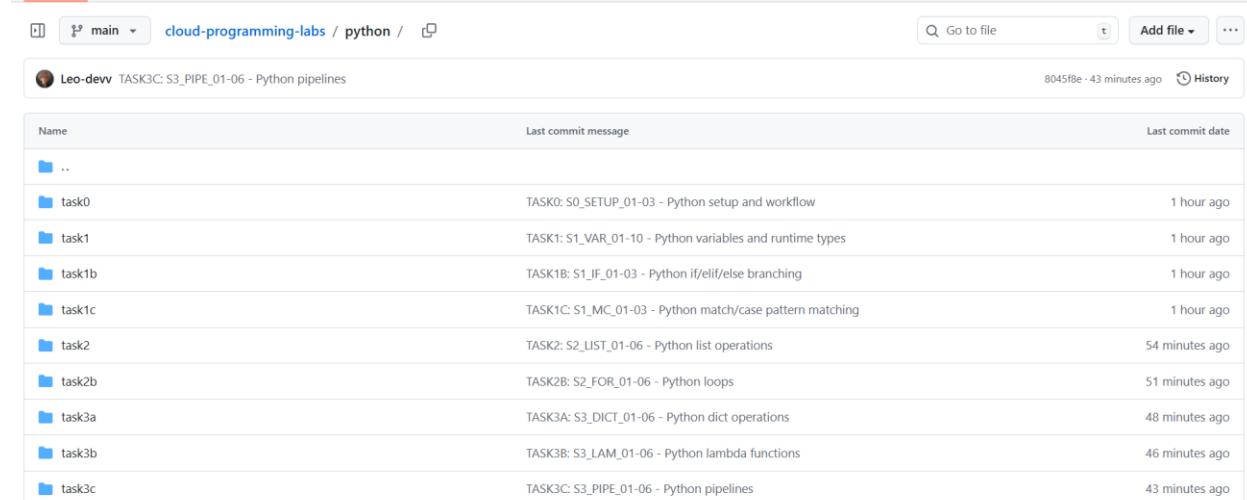
Project Overview

This report documents 50 Python exercises covering Python fundamentals (Python 3.10+), organized into 9 tasks including setup and optional bonus.

Repository Structure

The project structure follows course requirements:

```
python/ ┌── task0/      Setup (3 files) ┌── task1/      Variables and Types
      (10 files) ┌── task1b/     Conditional Statements (3 files) ┌── task1c/
      Match/Case Statements (3 files) ┌── task2/     List Operations (6 files) ┌──
      task2b/     Loop Constructs (6 files) ┌── task3a/     Dict Operations (6
      files) ┌── task3b/     Lambda Functions (6 files) ┌── task3c/     Pipelines
      (7 files including bonus)
```



The screenshot shows a GitHub repository named 'cloud-programming-labs' with a single branch 'main'. The repository contains a directory 'python' which is further divided into nine tasks: task0, task1, task1b, task1c, task2, task2b, task3a, task3b, and task3c. Each task folder contains several files corresponding to the structure described in the text above. The commit history for the 'main' branch shows 10 commits, each associated with a specific task and its purpose. The commits were made by 'Leo-devv' and occurred between 43 minutes ago and 1 hour ago.

Name	Last commit message	Last commit date
..		
task0	TASK0: S0_SETUP_01-03 - Python setup and workflow	1 hour ago
task1	TASK1: S1_VAR_01-10 - Python variables and runtime types	1 hour ago
task1b	TASK1B: S1_IF_01-03 - Python if/elif/else branching	1 hour ago
task1c	TASK1C: S1_MC_01-03 - Python match/case pattern matching	1 hour ago
task2	TASK2: S2_LIST_01-06 - Python list operations	54 minutes ago
task2b	TASK2B: S2_FOR_01-06 - Python loops	51 minutes ago
task3a	TASK3A: S3_DICT_01-06 - Python dict operations	48 minutes ago
task3b	TASK3B: S3_LAM_01-06 - Python lambda functions	46 minutes ago
task3c	TASK3C: S3_PIPE_01-06 - Python pipelines	43 minutes ago

Files follow the naming convention S[stage]_[type]_[number].py.

Task 0: Setup and Workflow

File	Description
S0_SETUP_01.py	Create project and print "It works"
S0_SETUP_02.py	File structure setup

S0_SETUP_03.py	Minimal test helper eq()
----------------	--------------------------

Task 1: Variables and Runtime Types

File	Description
S1_VAR_01.py	Catalog of values and type()
S1_VAR_02.py	Rebinding and dynamic typing
S1_VAR_03.py	Mutability: list vs tuple
S1_VAR_04.py	Identity vs equality (is vs ==)
S1_VAR_05.py	Truthiness
S1_VAR_06.py	Safe conversion: int() / float()
S1_VAR_07.py	NaN and math.isnan
S1_VAR_08.py	Big integers
S1_VAR_09.py	Type hints are not runtime enforcement
S1_VAR_10.py	Mini inspector

Task 1B: if/elif/else Branching

File	Description
S1_IF_01.py	Shipping cost with validation
S1_IF_02.py	Score to grade (A-F)
S1_IF_03.py	Normalize user name

Task 1C: match/case Pattern Matching (Python 3.10+)

File	Description
S1_MC_01.py	Day name using match/case
S1_MC_02.py	Command router using match/case
S1_MC_03.py	Simple calculator using match/case

Task 2: List Operations

File	Description
------	-------------

S2_LIST_01.py	Clean number strings
S2_LIST_02.py	Deduplicate without set
S2_LIST_03.py	Chunk a list
S2_LIST_04.py	Flatten one level
S2_LIST_05.py	Stats (min, max, avg, sum)
S2_LIST_06.py	Transform records

Task 2B: Loop Constructs

File	Description
S2_FOR_01.py	FizzBuzz+
S2_FOR_02.py	Find first even
S2_FOR_03.py	Sum until threshold
S2_FOR_04.py	Count occurrences (histogram)
S2_FOR_05.py	10x10 multiplication table
S2_FOR_06.py	Sum nested lists (matrix)

Task 3A: Dict Operations

File	Description
S3_DICT_01.py	Safe get by dotted path
S3_DICT_02.py	Merge defaults (shallow)
S3_DICT_03.py	Pick keys
S3_DICT_04.py	Omit keys
S3_DICT_05.py	Invert with collisions
S3_DICT_06.py	Group by property

Task 3B: Lambda Functions

File	Description
S3_LAM_01.py	Convert to lambdas
S3_LAM_02.py	Sort by key
S3_LAM_03.py	Closure factory (make_adder)
S3_LAM_04.py	Sum of squares of even numbers
S3_LAM_05.py	Higher-order predicate (at_least)
S3_LAM_06.py	Map values in a dict

Task 3C: Pipelines

File	Description
S3_PIPE_01.py	Implement pipe()
S3_PIPE_02.py	Implement compose()
S3_PIPE_03.py	String normalization pipeline
S3_PIPE_04.py	Iterable pipeline (generator-based)
S3_PIPE_05.py	Log line pipeline
S3_PIPE_06.py	Safe pipeline (pipe_safe)
S3_PIPE_07.py	Bonus: itertools (islice, groupby) + memory explanation

Git Workflow

Commit History

Development progressed through incremental commits:

Commit	Message
f8276e2	TASK0: S0_SETUP_01-03 - Python setup and workflow
bdb4e30	TASK1: S1_VAR_01-10 - Python variables and runtime types
d5f5252	TASK1B: S1_IF_01-03 - Python if/elif/else branching
acc9fa8	TASK1C: S1_MC_01-03 - Python match/case pattern matching
61cd5b1	TASK2: S2_LIST_01-06 - Python list operations
24bfdbc	TASK2B: S2_FOR_01-06 - Python loops
3ea4280	TASK3A: S3_DICT_01-06 - Python dict operations
f9c3d00	TASK3B: S3_LAM_01-06 - Python lambda functions
8045f8e	TASK3C: S3_PIPE_01-07 - Python pipelines

Tags

Tag	Commit
py-task0-done	f8276e2
py-task1-done	bdb4e30
py-task1b-done	d5f5252
py-task1c-done	acc9fa8
py-task2-done	61cd5b1
py-task2b-done	24bfdbc
py-task3a-done	3ea4280
py-task3b-done	f9c3d00
py-task3c-done	8045f8e
py-all-done	8045f8e

Testing

All exercises were tested using Python 3.10+. Each file contains executable code with output verification.

To run any exercise:

```
python python/task1/S1_VAR_01.py
```

The screenshot shows a code editor interface with several tabs at the top: .gitignore, JS S3_PIPE_06.js (which is active), and Report_58301_Javascript.pdf. The left sidebar shows a project structure with CLOUD PROGRAM..., js, python, task0, task1, and S1_VAR_01.py. The main area displays the contents of S1_VAR_01.py. Below the code editor is a terminal window showing the command: C:\Users\Rea\Downloads\Cloud programming Labs\python\task1\S1_VAR_01.py. To the right of the terminal is a table showing variable names, values, and types. The table has a header row: Name, Value, type(x), and type(x).__name__. The data rows are as follows:

Name	Value	type(x)	type(x).__name__
e__			
my_int	42	<class 'int'>	int
my_float	3.14	<class 'float'>	float
my_str	'hello'	<class 'str'>	str
my_bool	True	<class 'bool'>	bool
my_none	None	<class 'NoneType'>	NoneType
my_list	[1, 2, 3]	<class 'list'>	list
my_tuple	(1, 2, 3)	<class 'tuple'>	tuple
my_dict	{'a': 1}	<class 'dict'>	dict
my_set	{1, 2, 3}	<class 'set'>	set
my_func	<function>	<class 'function'>	function

The repository can be tested in Replit using the link above.

Screenshots

Extra Task

```

_PIPE_07.py
Logs grouped by level:
 DEBUG: 1 entries
 ERROR: 2 entries
 INFO: 4 entries

First 3 logs using islice:
 {'level': 'INFO', 'msg': 'Started'}
 {'level': 'INFO', 'msg': 'Processing'}
 {'level': 'ERROR', 'msg': 'Failed'}

Items 100-105 from infinite generator:
 [100, 101, 102, 103, 104, 105]

```

Replit Interface

The screenshot shows the Replit interface with several tabs open:

- Workflows**: Shows a list of workflows.
- cloud-programming-labs**: The active workspace.
- Uploads**: File upload section.
- Preview**: Preview section.
- Console**: Log output area.
- Git**: Git integration.
- Publishing**: Publishing options.

The **Console** tab displays the output of various Python scripts:

```

Running: python/task1b/S1_IF_01.py
shipping_cost(0.5, False): 5.0
shipping_cost(0.5, True): 4.5
shipping_cost(3, False): 10.0
shipping_cost(7, True): 18.0
shipping_cost(15, False): 50.0
shipping_cost(0, True): None
shipping_cost(-1, False): None

Running: python/task1b/S1_IF_02.py
grade(95): A
grade(85): B
grade(75): C
grade(65): D
grade(55): F
grade(0): F
grade(100): A
grade(-5): None
grade(101): None

Running: python/task1b/S1_IF_03.py
normalize_name(' Leo '): 'Leo'
normalize_name(''): 'Guest'
normalize_name(None): 'Guest'
normalize_name(' '): 'Guest'
normalize_name(123): 'Guest'
normalize_name('Alice'): 'Alice'

Running: python/task1c/S1_SW_01.py
day_name(0): None
day_name(1): Monday
day_name(2): Tuesday
day_name(3): Wednesday
day_name(4): Thursday
day_name(5): Friday
day_name(6): Saturday
day_name(7): Sunday
day_name(8): None

Running: python/task1c/S1_SW_02.py
System starting...
System stopping...
System restarting...
System is running
Available: start, stop, restart, status
Unknown command: foo

Running: python/task1c/S1_SW_03.py
calc(10, '+', 5): 15
calc(10, '*', 5)
calc(10, '+', 5): 50
calc(10, '/', 5): 2.0
calc(10, '/', 0): None
calc(10, '%', 5): None

Running: python/task1c/S1_VAR_01.py
Variable      Value          type()
user_name     'Leo'          str
user_age      25             int
is_active     True            bool

```

Git Tags

cloud-programming-labs Public

main 1 Branch 19 Tags

Go to file Add file Code

Leo-devv TASK3C: S3_PIPE_01-06 - Python pipelines 8045f8e · 45 minutes ago 15 Commits

File	Description	Time Ago
js	TASK3: S3_OBJ, S3_FN, S3_PIPE - objects, arrows, and pipelines	3 hours ago
python	TASK3C: S3_PIPE_01-06 - Python pipelines	45 minutes ago
.gitignore	TASK1: S1_VAR_01 to S1_VAR_10 - variables, types, scope, an...	6 hours ago
README.md	Initial setup: repository structure for JS exercises	7 hours ago
Report_58301_Javascript.pdf	Add report PDF	2 hours ago

53 minutes ago → 24bfbdbc zip tar.gz

py-task2-done

1 hour ago → 61cd5b1 zip tar.gz

py-task1c-done

1 hour ago → acc9fa8 zip tar.gz

py-task1b-done

1 hour ago → d5f5252 zip tar.gz

py-task1-done

1 hour ago → bdb4e30 zip tar.gz

py-task0-done

1 hour ago → f8276ee2 zip tar.gz

py-all-done

45 minutes ago → 8045f8e zip tar.gz

all-done

3 hours ago → f9c44a6 zip tar.gz

The screenshot shows a GitHub repository named 'cloud-programming-labs'. It has 1 branch and 19 tags. The main branch contains files for 'js' and 'python' tasks, along with '.gitignore', 'README.md', and a 'Report_58301_Javascript.pdf'. A commit from 'Leo-devv' titled 'TASK3C: S3_PIPE_01-06 - Python pipelines' was made 45 minutes ago. Below the commits is a list of task completion markers: 'py-task2-done', 'py-task1c-done', 'py-task1b-done', 'py-task1-done', 'py-task0-done', 'py-all-done', and 'all-done'. Each marker is followed by a timestamp, a commit hash, and download links for 'zip' and 'tar.gz' formats.

Summary

All pyexercises are complete and tested (Task 0: 3 files, Task 1: 10 files, Task 1B: 3 files, Task 1C: 3 files, Task 2: 6 files, Task 2B: 6 files, Task 3A: 6 files, Task 3B: 6 files, Task 3C: 7 files including bonus). The project follows course guidelines for structure, file naming, and version control.