1. Що таке Git та для чого використовується? Як створити новий локальний репозиторій Git? Наведіть команди.

Git-це система контролю версій.

Для створення нового репозиторію потрібна відкрити головну сторінку та вибрати на ній Створення нового репозиторію. Після у відкритому вікні обрати ім’я для нього, відкритий чи приатний доступ, та поставити прапор на инициализування файлу Readme

$ git add- додаванняя файлу до репозиторію

## $ git status- відоброзити статус репозиторію

1. Наведіть команди git для додавання файлів та директорій ./dev ./dev/file.c ./.gitignore до локального репозиторію одним комітом (уважно).

Для тогго, щоб додати вайли вікористовується команда git add з обраним до нас типом файлів .

$ git add .gitignore

1. Перерахуйте та коротко окресліть відомі вам **базові** типи даних Python.

Числа- числа , що записуеться до змінних (бувають цілими та з плавуючею комою )

Словарі- невпорядочена структура об’єктів з данним їм ключем

Список- упорядкована структура об’єктів ріних типів

Строки- незммінна послідовність символів( можна за допомогою індексіів витягати нам потрібні значення )

Булеві значення – змінні які мають два типи значень(True False)

Mножества – стуктура у якій є неупорядковані, неіндексовані елементи

Кортежи – Структура як список, але з незмінною структурою даних

1. Числові типи Python. **Коротко** опишіть з прикладами використання. Створіть функцію, що розраховує реактивний опір конденсатора з ємністю *C* [Ф] на частоті *f* [Гц]. Якщо значення *f* не задано користувачем, за замовчуванням використати 1 кГц.

Цілі числа , комплексні .з фіксовано точністю, множини та логічні значення

C=Input()

F=Input()

If F is None:

F=1000

Print=(1/(6.18\*C\*F))

1. Створіть функцію, яка приймає в якості першого аргументу бажаний опір паралельно з'єднаних резисторів *R* [Ом], а в якості подальших аргументів – величини резисторів, що є в наявності [Ом]. Функція повертає кортеж (tuple), першим елементом якого є словник типу {номінал: кількість}, а другим елементом – абсолютне відхилення результуючого опору від бажаного [Ом].

Def asd():

R=Input()

Res\_In=[]

Elem=0

while Res\_In.append(Elem=elem+input()) not None:

R=R-elem

T={ [Res\_In] : Res\_In.len(), R}

return T

1. Наведіть приклад функції, що **коректно** приймає в якості значення за замовчанням Mutable-об'єкт. Чому коректно робити саме так?

|  |
| --- |
| list = [10, 20, 30]  list[0] = 40  print(list) |

Викорестаня такого обьекта дозволяє нам заміняти данні при не зміні індефікатору. Окрім цього незмінні обьекти в памяті набагато більщі і для іх зміни потрібно ініціалізувати окремий доступ.

1. Тернарний оператор Python. Навести код, що **за допомогою тернарного оператора** присвоїть змінній var значення 69 якщо змінна arg задана в None, в іншому випадку – присвоїть значення arg. Як зробити це без тернарного оператора? Наведіть приклад коду.

Def asd(var, arg):

Return var=arg if arg not None else 69 : - тернарний оператор

Def asd(var,arg):

If arg not None:

Return var=arg

Return arg=69

1. Які цикли Python вам відомі? Яким чином можна зімітувати функціонал циклу do-while з мови С в Python?

Цикли for та while .

While – цикл який буде виконуватися доки умова виконання істина

For – буде виконуатися , поки ітеруємий об’єкт не буде повністю переборен (ітерован)

Зімітувати цикл do-while у пітоні мона інізіалізувавши функцію перед вайлом і надати потрібні після вимогу.

Funk()

While not fail\_funk

Funk()

Або створивши потрібну нам змінну та надати їй булевий формат

Funk = True

while Funk:

Funk = test\_funk()