

Домашнее задание по теме «Введение в Python. Основные конструкции и операторы. Репозитории»

Формулировка задания

1. Установить python на локальный компьютер.
2. Установить IDE для работы на локальный компьютер с расширением под python и работу с Jupiter Notebook.
3. Создать локальный и удаленный репозиторий.
4. Решить две небольшие задачи на python. Сохранить задачи (сделать коммиты для каждой) в локальном git и опубликовать в удаленном репозитории.

Описание плана работы

Задание 1 Установить python на локальный компьютер

Разверните рабочую среду для разработки на python на вашем компьютере. Для этого необходимо установить Python и библиотеку notebook для работы с Jupiter Notebook.

Убедиться, что python установлен для всех пользователей и открывается в консоли компьютера.

Сделайте 2 снимка экрана для отчета по работе:

1. Проверка запуска python в командной строке.
2. Проверка запуска Jupiter Notebook в браузере.

Задание 2 Установить IDE

Скачать и установить актуальную среду разработки (IDE) для python.

Выбрать один из вариантов:

- 1) VS Code. URL: <https://code.visualstudio.com/>
- 2) GigaIDE (основана на PyCharm IDE). URL: <https://gitverse.ru/features/gigaide/>
- 3) GitHub Desktop (основана на VS Code IDE). URL: <https://desktop.github.com/download/>

В среде разработки настроить расширения для работы на python. Установить виртуальную среду.

Открыть в среде разработки блокнот Jupiter Notebook с лекции и запустить.

Для отчета по работе сделать снимок экрана с запущенным Jupiter Notebook в IDE.

Задание 3 Создать репозиторий

Установить git на локальный компьютер. Инструкция: <https://github.com/git-guides/install-git>.

Зарегистрируйтесь и создайте репозиторий для дисциплины на GitHub (<https://github.com>) или аналогах.

Склонируйте созданный репозиторий на локальный компьютер.

Для отчета по работе сделать снимок экрана с созданным удаленным репозиторием.

Задание 4 Решить простые задачи с основными конструкциями и операторами

Решить следующие задачи на python на локальном компьютере. Решение представить в формате .ipynb или .py. Сохранить задачи (сделать коммиты для каждой) в локальном git и опубликовать в удаленном репозитории.

Задача 1. Максимум, минимум, остальное

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число. На вход программы могут подаваться и повторяющиеся числа.

Примечание: Дополнительная информация по функции print(). Следующая строка

`print(3, 1, 2, sep="\n")` выведет числа каждое на следующей строке. Подробности про свойство sep по URL: https://www.w3schools.com/python/ref_func_print.asp

Вход программы 1:

8
2
14

Выход программы 1:

14
2
8

Вход программы 2:

23
23
21

Выход программы 2:

23

21

23

Для отчета по работе выполнить задание в файле `irunb` или `ru`. Сделать снимок корректного выполнения программы в IDE.

Задача 2. Счастливый билет

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета. Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подается строка из шести цифр. Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Вход программы 1:

090234

Выход программы 1:

Счастливый

Вход программы 2:

123456

Выход программы 2:

Обычный

Для отчета по работе выполнить задание в файле `irunb` или `ru`. Сделать снимок корректного выполнения программы в IDE.

Перечень необходимых инструментов

- Python
- Jupiter Notebook
- IDE VS Code
- GigaIDE

Форма предоставления результата

1. В поле ссылки загрузить ссылку на удаленный репозиторий с доступом для наставника.
2. В поле файла загрузить архив с папкой, в которой разместить отчет со скриншотами по заданиям и решение задачи 1 и 2. Решения должны быть представлены в формате .ipynb или .py.

Шкала оценивания

- 1.0 – отлично
- 0.7–0.9 – хорошо
- 0.5–0.6 – удовлетворительно
- Менее 0.5 – задание не выполнено