



Отчет о проверке

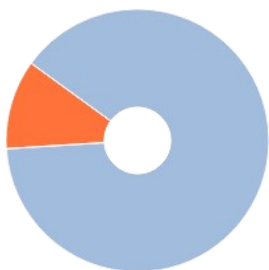
Автор: ApiCorp

Название документа: Kulakov Denis Report(1).pdf

Проверяющий: ApiCorp

Организация: Национальный Исследовательский Университет "Высшая Школа Экономики"

РЕЗУЛЬТАТЫ ПРОВЕРКИ



Совпадения:
10,6%



Оригинальность:
89,4%



Цитирования:
0%



Самоцитирования:
0%



1

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует проверенному тексту документа.

- **Совпадения** — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» – это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.
- **Самоцитирования** — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» – это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.
- **Цитирования** — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» – это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.
- **Текстовое пересечение** — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
- **Источник** — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
- **Оригинальный текст** — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» – это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Номер документа: 805109

Тип документа: Курсовая работа

Дата проверки: 02.06.2024 22:12:01

Дата корректировки: Нет

Количество страниц: 21

Символов в тексте: 15045

Слов в тексте: 2206

Число предложений: 112

Комментарий: не указано

ПАРАМЕТРЫ ПРОВЕРКИ

Выполнена проверка с учетом редактирования: Да

Выполнено распознавание текста (OCR): Нет

Выполнена проверка с учетом структуры: Нет

Модули поиска: Цитирование, Библиография, Публикации РГБ, Перефразирования по СПС ГАРАНТ: аналитика, Диссертации НББ, Переводные заимствования*, Шаблонные фразы, ИПС Адилет, СПС ГАРАНТ: нормативно-правовая документация, Издательство Wiley, Коллекция НБУ, Переводные заимствования (RuEn), СПС ГАРАНТ: аналитика, Патенты СССР, РФ, СНГ, Кольцо вузов, Публикации eLIBRARY, Перефразирования по Интернету, Перефразирования по коллекции издательства Wiley, Перефразирования по коллекции IEEE, Сводная коллекция ЭБС, Переводные заимствования IEEE, Переводные заимствования издательства Wiley, Перефразированные заимствования по коллекции Интернет в английском сегменте, Публикации eLIBRARY (переводы и перефразирования), Перефразированные заимствования по коллекции Интернет в русском сегменте, СМИ России и СНГ, Перефразирования по Интернету (EN), Переводные заимствования по Интернету (EnRu), Медицина, Кольцо вузов (перефразирования), Переводные заимствования по коллекции Интернет в английском сегменте, Интернет Плюс*, IEEE, Переводные заимствования по коллекции Интернет в русском сегменте, Переводные заимствования по коллекции Гарант: аналитика, Собственная коллекция компании

ИСТОЧНИКИ

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарий
[01]	3,14%	3,14%	Bozhenov O.D. Project Report.pdf	07 Июн 2023	Собственная коллекция компании	
[02]	2,52%	2,52%	Баранков В.Л., Волкова Н.С., Дми... http://ivo.garant.ru	02 Июл 2016	Переводные заимствования по коллекции Гарант: аналитика	
[03]	2,42%	0,54%	Software project AMI - Report, Titl...	07 Июн 2023	Собственная коллекция компании	
[04]	1,83%	0%	Coursework_report_Zheng_Yuxin-...	13 Мая 2024	Собственная коллекция компании	
[05]	1,83%	0%	fishman_m_d_bot-analitik.pdf	29 Мая 2021	Собственная коллекция компании	
[06]	1,75%	0%	Containerized Architecture for Ed... https://ieeexplore.ieee.org	01 Июн 2020	IEEE	
[07]	1,73%	1,73%	Docker Swarm Orchestration for H... https://ieeexplore.ieee.org	21 Мар 2024	Перефразирования по коллекции IEEE	
[08]	1,54%	0,57%	http://archiv.ub.uni-heidelberg.de... http://archiv.ub.uni-heidelberg.de	10 Фев 2023	Интернет Плюс*	
[09]	1,3%	0%	Title page (Software project) team ...	09 Июн 2023	Собственная коллекция компании	
[10]	1,26%	0,05%	Том 33, № 3 (2021) https://ispranproceedings.elpub.ru	30 Дек 2022	Интернет Плюс*	
[11]	1,19%	0,23%	https://www.eurchembull.com/.u... https://eurchembull.com	30 Мая 2024	Интернет Плюс*	
[12]	1,19%	0%	https://ebuah.uah.es/dspace/bitst... https://ebuah.uah.es	26 Янв 2023	Интернет Плюс*	
[13]	1,19%	0%	https://www.hse.ru/data/2022/05/... https://hse.ru	05 Июн 2023	Интернет Плюс*	
[14]	1,19%	0%	CourseProjectGame.pdf	01 Июн 2024	Собственная коллекция компании	
[15]	1,19%	0%	KOLCHIN-final-report.pdf	29 Мая 2024	Собственная коллекция компании	
[16]	1,19%	0%	Coursework_Kozhevnikov_2023_BP...	06 Июн 2023	Собственная коллекция компании	
[17]	1,11%	0,54%	Collaborative Visual Programming... https://ieeexplore.ieee.org	23 Июн 2023	IEEE	
[18]	1,1%	1,1%	Flight Software Dictionary Develop... https://ieeexplore.ieee.org	10 Авг 2022	IEEE	
[19]	1,07%	0,19%	Difference Between A Docker Ima... https://redswitches.com	30 Мая 2024	Интернет Плюс*	
[20]	0,96%	0%	Docker Swarm Orchestration for H... https://ieeexplore.ieee.org	21 Мар 2024	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[21]	0,91%	0%	https://repositorio-aberto.up.pt/bi... https://repositorio-aberto.up.pt	02 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[22]	0,91%	0%	Optimal Resource Provisioning for... https://shodhganga.inflibnet.ac.in	02 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[23]	0,81%	0%	Considerations for Achieving Cros... https://frontiersin.org	04 Ноя 2020	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.

[24]	0,81%	0%	Evaluation of File Carving Tools fo... https://ieeexplore.ieee.org	04 Янв 2023	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[25]	0,75%	0%	What's the difference between Vir... https://scalecomputing.com	30 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[26]	0,69%	0%	Management and Quality Control ... https://frontiersin.org	01 Фев 2022	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[27]	0,69%	0%	Efficient Model-Data Integration f... https://frontiersin.org	21 Мая 2020	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[28]	0,69%	0%	Integration of "omics" Data and Ph... https://frontiersin.org	30 Мая 2020	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[29]	0,69%	0%	Frontiers Blockchain Technology... https://translated.turbopages.org	15 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[30]	0,68%	0%	Containerized MPI Application on ... https://ieeexplore.ieee.org	15 Июл 2022	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[31]	0,66%	0%	2016ВКР030341МАРИН.docx	27 Мая 2016	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[32]	0,66%	0%	Б-9-422	06 Апр 2024	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[33]	0,62%	0%	РАЗРАБОТКА СИСТЕМЫ ДЛЯ ОПР...	16 Июн 2017	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[34]	0,57%	0%	СОВМЕСТНОЕ ДИСТАНЦИОННОЕ... https://elibrary.ru	31 Дек 2021	Публикации eLIBRARY	Источник исключен. Причина: Маленький процент пересечения.
[35]	0,56%	0%	skvorcov_i_a_servis-interpretacii-p...	27 Июн 2023	Собственная коллекция компании	Источник исключен. Причина: Маленький процент пересечения.
[36]	0,55%	0%	jitniy_g_d_portativnyy-blokcheyn-d...	24 Мая 2024	Собственная коллекция компании	Источник исключен. Причина: Маленький процент пересечения.
[37]	0,54%	0%	How to Improve Docker Container ... https://squash.io	20 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[38]	0,52%	0%	PlanoTse Handbook for Job Search... https://en.wikibooks.org	02 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[39]	0,47%	0%	Pharmacology http://studentlibrary.ru	16 Янв 2018	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[40]	0,46%	0%	ДИПЛОМ Мацагов Джабраил.doc	27 Июн 2013	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[41]	0,44%	0%	АННОТАЦИЯ К: Актуальные проб... http://ivo.garant.ru	05 Сен 2022	СПС ГАРАНТ: аналитика	Источник исключен. Причина: Маленький процент пересечения.
[42]	0,44%	0%	ВЫБОР НАИЛУЧШЕЙ ЭНЕРГОСБЕ... https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[43]	0,44%	0%	Гражданское право и его роль в ... https://e.lanbook.com	21 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[44]	0,44%	0%	Инновационное развитие эконо... https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[45]	0,31%	0%	http://www.kgau.ru/new/all/scien... http://kgau.ru	04 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.


NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Applied Mathematics and Informatics"

Software Project Report on the Topic:
Development of Materials for Teaching Programming in Python

Fulfilled by:

Student of the Group
БПА/212
Kulakov Denis Alexeevich



(signature)

May 31st, 2024
(date)

Checked by the Project Supervisor:

Voznesenskaya Tamara Vasilievna
Associate Professor
Big Data and Information Retrieval School, Faculty of CS, HSE University

(signature)

June 2nd, 2024
(date)

Moscow³ 2024

Table of contents

Abstract	3
Introduction	4
Subject area	5
Relevance	6
Main goal	8
Main tasks	8
Instruments	8
Basic terms and definitions	9
Project implementation	11
Existing alternatives and analogues: comparative analysis	18
Results and conclusion	19
Results description	19
Perspectives for future work	19
Conclusion	20
Sources	21

Abstract

This project aims at addressing the increasing demand for quality secondary-level computer science education. The goal is to ultimately create a complete Python curriculum that is accessible, free to students and open-source. The project uses contemporary software development applications like Docker, Git among others and is designed to ensure that students not only listen and learn the basics of programming but also gain an understanding of how things are really done in today's business climate. Students gain experience creating consistent development environments using containerization, Docker. Learn git: Version control and collaborative work are crucial in any software related work, and git makes you use version control along with a collaborative workflow.

The contents consist of slide decks, code examples, and self-evaluation resources to support an interactive and engaging learning process. It highlights practical examples of Python, showing it in action in all sorts of real-world situations to encourage and motivate students. This project stands out, compared to the current educational programs, for the integration of modern development practices and is stronger in a comparative analysis.

Introduction

The project, 'Developing material for the study of Python programming,' is intended to produce teaching and learning material focused on the teaching of Python programming to secondary school students. Python has become a language of choice for teaching introductory computer science in many places because of its cleanness and flexibility for teaching both basic programming constructs and complex applications. In the end of this description, you will find a recent article on the project's desire to create complete and pedagogically rich learning objects that are neither text-based nor text-centric, meeting the demands of the contemporary academic environment.

The course introduces students to modern industry standards and practices using the contemporary software development tools; namely Docker and Git. In addition, to Docker, we remove the various local development system configuration disparities by providing lightweight and self-contained computing environments. Students learn distributed version control with Git, which is a prelude to collaborative and iterative development workflows leveraged in the software industry. This project is a master class to develop an expert understanding and create sophisticated programming concepts based on basics that is directed to more advanced studies for students during their learning experiences.

The end game, is a true educational supplement, something that not only learns your tech, but give you tools to tame the tangled world of modern computational problem-solving. The project aims at systematically considering this goal as an evidence-based aspect of what computer science education contributes to a technology workforce pipeline.

Subject area

The goal of this project is to raise public awareness to computer science education and its emphasis on:

Contemporary software development tools:

- **Docker** - Explains the concepts of containerization, and teaches students to create applications in development environments.
- **Git** - Version control, collaborative workflow, and versioning in developing software.

Real-world applications:

- **Practical Python Programming**, which shows students what they can do with Python and how to use it for all types of real-world scenarios.

Relevance

For a number of reasons, the project has implications that are all too significant in the educational and technological landscape of today.

1. Wide application

- 1.1. Python has broad application as well, for applications from internet development and automation to data analysis and machine learning. Because of this, learning it can be very useful to students learning it can help in applying it differently.
- 1.2. Its wide use in professional and educational settings helps provide a robust community and plethora of learning resources.

2. Industry demand

- 2.1. Programming skills are in high demand, from tech to finance and healthcare, and many other fields. And python specifically is used very frequently because of how robust it is and how easy it is to learn.
- 2.2. Career opportunities.

3. Educational needs

- 3.1. Programming is becoming more widespread in the curricula of schools and educational institutions in order to equip students with skills that are highly valued in the job market of the future
- 3.2. Python has a syntax that is simple and easy to read, making it suitable for beginners to understand most programming concepts. This reduces the barrier to learn and open the door for more people to learn programming.

4. Integration of modern development practices

- 4.1. **Docker usage:** Telling students to use Docker is a good way to make them understand what containers are and why they are important for modern software development. It is practicing what

we preach which is essential for real world application as it gives us practical experience working in creating consistent development environments. Docker is among the rising trends in the hiring market to follow down on — the availability of Docker as the must-have skill has become a joke with tips for getting a DevOps job.

- 4.2. **Git:** Teaching Git, but this falls in the line with the previous tip and this is just how you usually do things. GIT helps students to manage code changes, collaborate with others on projects and understand how changes are introduced and they track each change. Students going through an educational environment will appear more attractive on the job market when they already have the foundation with how to use Git as it gets used in the majority of development workflows out there today.

Main goal

It is a project aimed at providing basic understanding in programming on Python to school students and general know how of software development.

Main tasks

1. Go through the Materials Given by the Supervisor
2. Review analogues and alternatives to accessible analogues in existence
3. Create the courseware and the concept of the course.
4. Create the materials
5. Write the project report
6. Results

Instruments

1. Google Drive
2. Google Docs
3. Jupyter Notebook
4. Google Slides
5. Git
6. Docker
7. Google Collab
8. SlidesGo

Basic terms and definitions

Programming language – an artificial language designed to communicate and enforce calculations that may be performed by a tool, most generally a computer.

Python — A hopeful programming language that is high-level, interpreted mentioning in readability, and applicable with scripting and software development tasks.

Educational materials - required learning material, such as texts, presentations and interactive exercises.

Git - is a version control system for tracking changes in source code during software development.

Repository (repo) – where the project's files and history are stored.

Commit - this is a save of the work you have done on your project (i.e this is what your project looked like at a particular point in time)

Branch - A distinct line of development.

Branch Merge - Pull changes from various branch or repository.

Clone - This means copying an already existing repository.

Remote - Your repository that is hosted on the internet or a network

Staging area - Where your changes are staged before you commit.

Pull request (merge request) - A request to merge one branch of code into another (usually from a developer's branch into the main or development branches).

Rebasing - A sequence of commits can be moved or combined onto a new base commit with rebasing. It is mostly used for the synchronization of a feature branch with the main branch.

Cherry-picking - taking a commit from one branch and attaching to another.

Stashing - used to temporarily save you changes (not commit)

Docker - is complete ecosystem (platform as a service) based on the OS-level virtualization i.e., containers.

Image - A lightweight, standalone, executable package that includes everything needed to run a piece of software, including the code, a runtime, libraries, and dependencies

Container⁷ - A container is a running instance of an image.¹⁹

Docker file - Text file that includes a list of commands to build a Docker image.

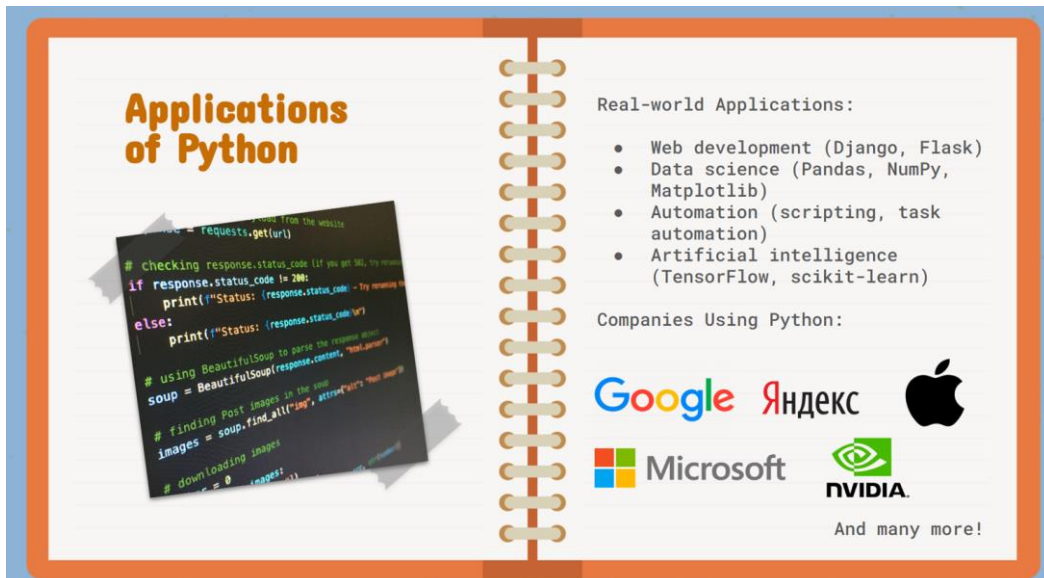
Docker hub¹¹ - a cloud-based registry in which Docker users and partners make use of Docker.

Project implementation

This collection of branches and stages of work is structured to construct a broad educational program which is efficient for secondary school students in the project. This list of tasks can be found below:

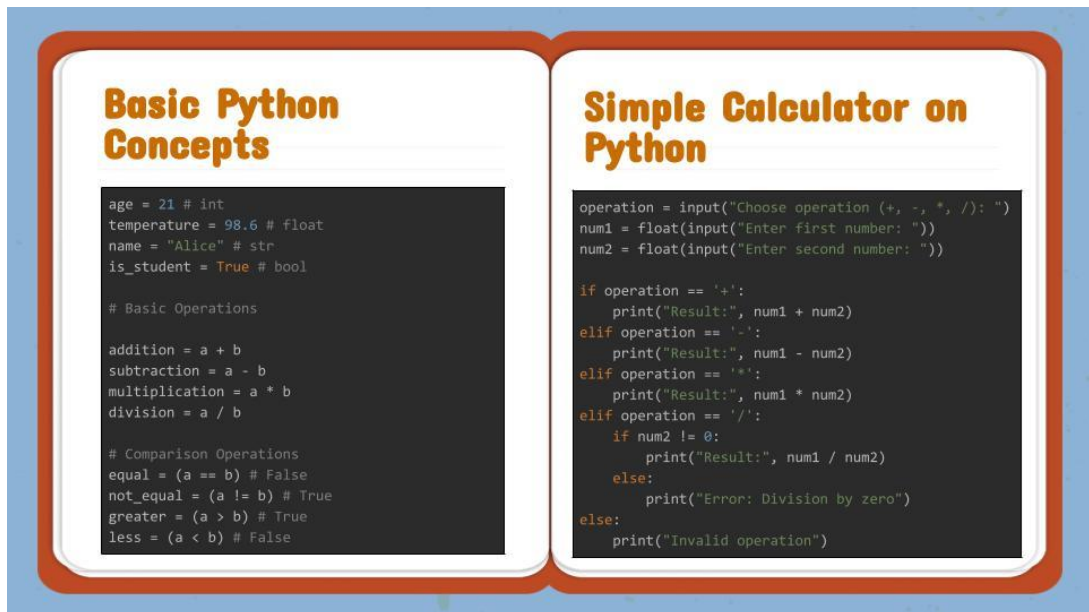
1. Master classes for school students:

1.1. Curriculum development: defining a list of Python programming concepts that could be made into master class topics.



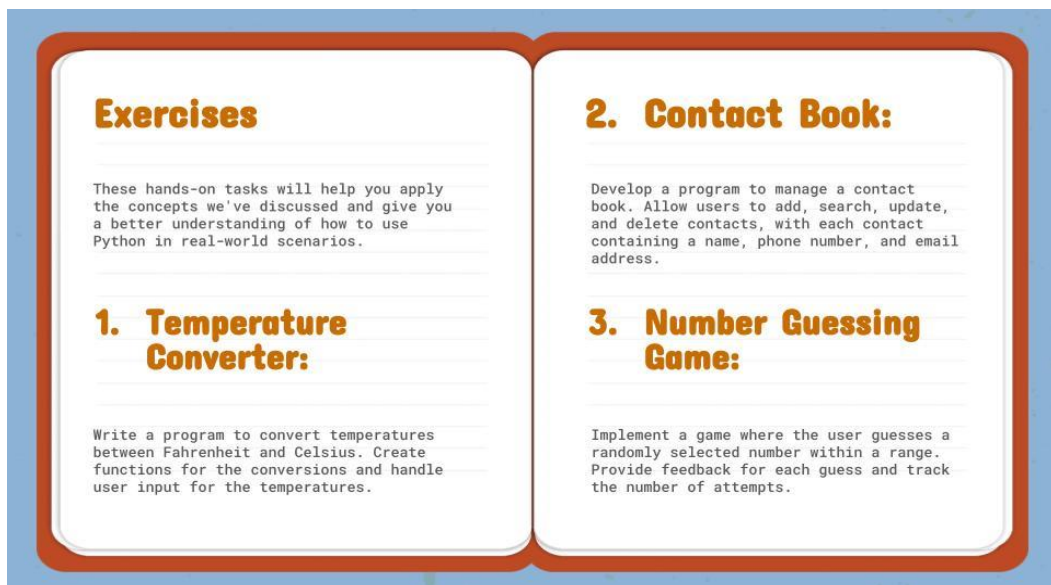
Picture 1. Real-world applications for Python

1.2.Creating resource: Create additional material such as slide decks and code examples.

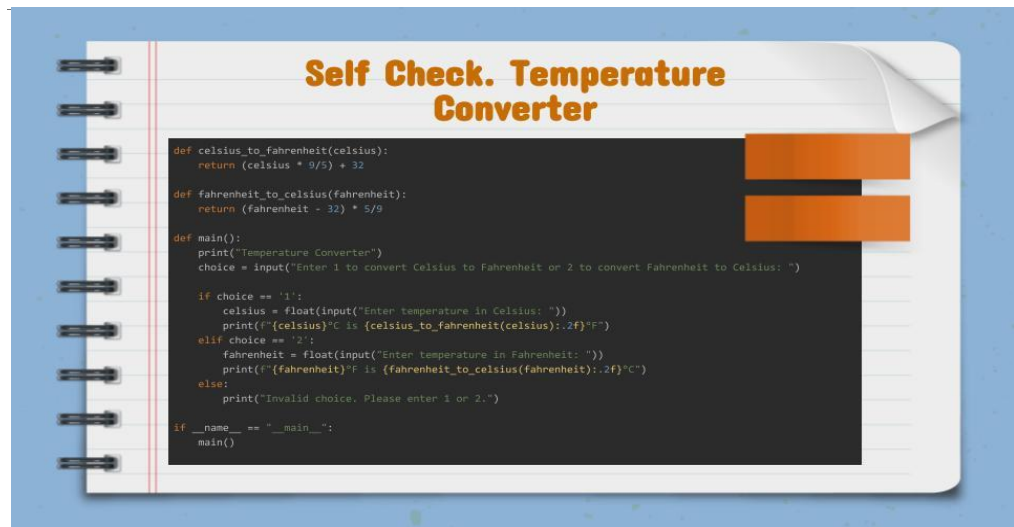


Picture 2. ² Code Snippets

1.3.Create self-assessment instruments to evaluate student understanding, and evaluate how effective the sessions are in the form of master classes.



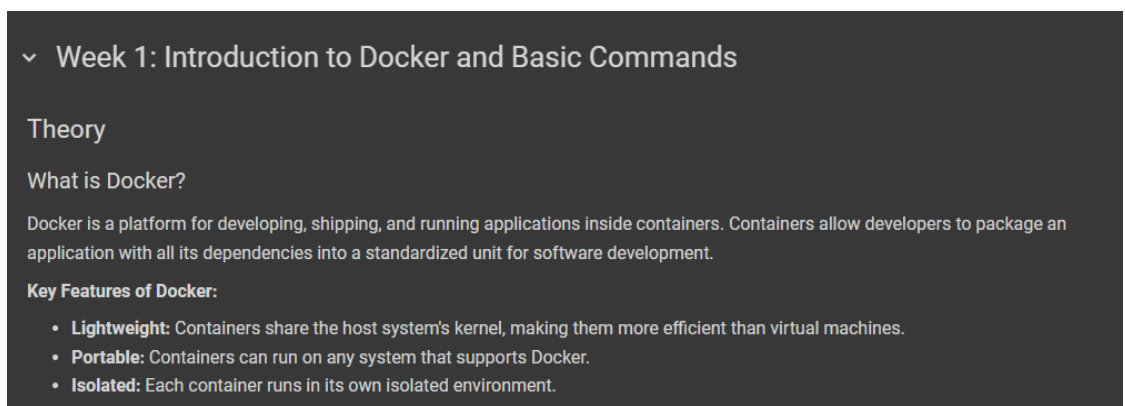
Picture 3. Exercises example



Picture 4. Solutions for self check

2. Containerization concepts with Docker:

2.1. Introducing containers, and why it is useful



Picture 5. Key features

2.2. Create a article tutorial on the setup and running of Docker Containers.

Installing Docker

To start using Docker, you need to install it on your machine.

On Ubuntu:

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

On macOS/Windows:

Download and install Docker Desktop from [website](#)

Picture 6. Installation guide

▼ Running a Container

```
# Run a simple container  
docker run hello-world
```

Example Usage: Use `docker run` to start a container from an image. This command downloads the hello-world image, if not already present, and runs it. This is a great way to test if your Docker installation is working correctly.

▼ Listing Containers

```
# List running containers  
docker ps  
  
# List all containers  
docker ps -a
```

Example Usage: Use `docker ps` to list running containers and `docker ps -a` to list all containers, including stopped ones. This helps you manage and see the status of your containers.

▼ Stopping and Removing Containers

```
# Stop a running container  
docker stop container_id  
  
# Remove a stopped container  
docker rm container_id
```

Picture 7. Management of docker container

- 2.3. Do practical exercises that lead student on how to build, run and manage Docker containers.

Homework

1. Create a Docker Compose File:
 - Write a `docker-compose.yml` file for a web application and a database service.
2. Run Multi-Container Application:
 - Use Docker Compose to start the multi-container application.
3. Manage Services:
 - List running services and stop the application using Docker Compose.
4. Explore Docker Compose Commands:
 - Experiment with additional Docker Compose commands like `docker-compose logs` to view the logs of all services, `docker-compose restart` to restart services, and `docker-compose build` to build or rebuild services.

Picture 8. Example of exercises

- 2.4. Integrate Docker learning modules into the Python curriculum to solidify the concept of containerizing software.

Example Dockerfile:

```
# Use an official Python runtime as a parent image
FROM python:3.8-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

Picture 9. Usage of Docker with Python parent image

3. Working with Git:

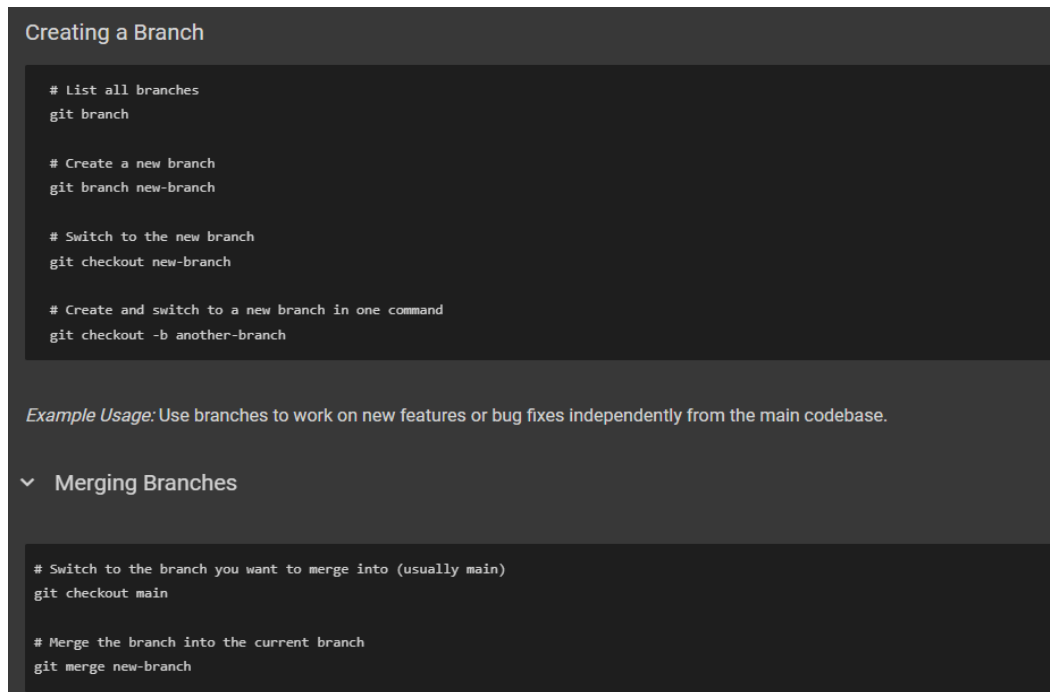
- 3.1. Prepare a list of starting points for Git commands such as cloning a remote repository, making changes, committing changes, and then pushing changes back to the remote repository.

Commands Recap:

```
git branch          # List all branches
git branch new-branch # Create a new branch
git checkout new-branch # Switch to a new branch
git checkout -b another-branch # Create and switch to a new branch
git merge new-branch  # Merge a branch
git branch -d feature-branch # Delete a branch
```

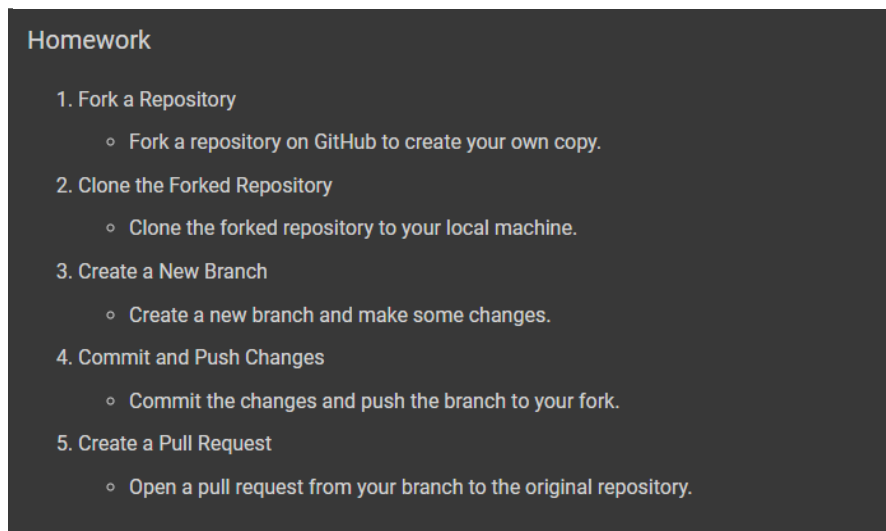
Picture 10. Recap of commands

- 3.2. Teach students what branching strategies are, how to manage a merge conflict correctly.



Picture 11. Branching commands example

3.3. Create tasks how to explore materials on collaboration with Git.



Picture 12. Exercises for collaboration

Existing alternatives and analogues: comparative analysis

In order to disclose the project positioning and a potential impact, let me briefly review the current state of education programs in Russia:

- 1) **Yandex Lyceum:** A 2-year python programming course for high school kids
- 2) **Kaspersky Cybersecurity for the Next Generation:** Cybersecurity and programming basics.
- 3) **Moscow Coding School:** Programming courses (Python) for all ages
- 4) **Codewards:** Deliver educational coder courses for school students in the form of games

While those programs are extensively theoretical and academic, this project is a practical one oriented at modern tooling hands-on to 21st century software development and version control. The materials serve as a guide and are meant to be modified according to the teaching technique and the educational environment.

In conclusion, printable educational materials that will be flexible and adaptable (for other teaching styles and educational needs) are created. As opposed to the very pre-programmed curricula the state or country has only some existing programs have.

Results and conclusion

Results description

A 4 weeks course on Git, 2 weeks course on Docker, and a Python workshop for high school students. This is the whole list (downloadable from shared git repository):

Perspectives for future work

This project has a few different perspectives for future work:

- 1) To build a website for education, exercise and training videos
- 2) Develop teacher training programs with the new curriculum
- 3) Create A Student and Teacher Community to share insights and collaborate on projects.

Conclusion

The purpose of the project “Developing Education Materials for Teaching Python Programming” is to deliver well-structured exhaustive educational materials especially for secondary school students so they can soon become productive. This is done by promoting Python as the simple and strong language of choice for the real world.

Focus on modern delivery — use of development tools like Docker and Git was seen during the project. We ensure students learn to utilise the same type of rigorous development settings that are in use at companies, such as Git for version control workflows, and Docker for creating consistent development environments, and do so at a distributable classroom level and the students walk away with a very marketable set of skills.

These resources are designed for you to be able to do interact, and can be used in a number of learning environments. It offers a hands-on experience with Python, and incorporates self-check questions to help students realize the worth of their learning efforts and ascertaining to what extent they are taking advantage of the course.

This project, which is useful in making a unique contribution to the others in terms of the integration of contemporary tools and the learning steps to be adaptable, hashed against other similar protocols. By offering something as is not currently available, this approach is not just filling gaps in what we offer today, it is also upping what the high school level of computer science can be.

Sources

1. Python Software Foundation. Python Programming Language. Retrieved from <https://www.python.org/>.
2. Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239), 2. 18
3. Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.
4. Yandex. Yandex.Lyceum. Retrieved from <https://lyceum.yandex.ru/>.
5. Kaspersky Lab. CyberSecurity for the Next Generation. Retrieved from <https://www.kaspersky.com/>.
6. Moscow Coding School. Programming Courses. Retrieved from <https://codingschool.moscow/>.
7. Codewards. Game-based Learning Platform. Retrieved from <https://www.codewards.com/>.
8. GitHub. The Git Distributed Version Control System. Retrieved from <https://github.com/>. 17
9. SlidesGo. Free Google Slides and PowerPoint Templates. Retrieved from <https://slidesgo.com/>.