

Python Homework Help

What did you learn during lessons lessons? to bot yourself. Besides the peda? goog analyzing the results of the execution homework, can reveal difficult?

stirs that a student faces and give extra time for better? mastering the "problem" material. how

Do parents usually formulate a problem?

mu as follows:

1. Does the child refuse to perform before? car task;
2. To perform tasks required too much time;
3. The child is constantly distracted;
4. The child can not engage in itself?

independent and needs constant the presence and assistance of an adult.

Python is considered a universal programming language, it is simple and understandable for a beginner, it is pleasant to write and work with professionals on it.

The scope of use of Python is quite wide - from web development to process automation, as well as data processing, including the currently popular DataScience and Machine Learning.

However, as with any other language, the development of syntax is not enough to begin its successful and effective application.

Our course will introduce you to the idioms of the language and teach them to practice.

FOR WHOM IS THIS COURSE

The course is designed for participants with basic knowledge of the basics of programming, who are interested in getting acquainted with Python and further applying it in various areas of development.

AT THE END OF COURSE YOU

Get Python programming skills

Learn how to create and test applications in [Python](#)

You will be able to start exploring trend areas in development, such as Data Science and Machine Learning

The course is designed for two and a half months and consists of 11 lessons. The duration of each lesson is 4 hours, on Saturdays.

One lesson consists of 2 hours of theoretical block and 2 hours of practical block. After each class homework is issued.

The course covers the following topics:

Introduction to Python

- The history of the language
- Basic operators, data types
- Language interpreters
- Basic operators, data types
- Functions

- Packing, unpacking arguments and assignment operator, key arguments and default arguments
- Scopes, global and nonlocal operators
- Functional programming, anonymous functions, map, filter and zip
- Generators lists, sets and dictionaries
- Decorators
- Decorators Syntax
- Decorators with and without arguments
- Examples of the use of decorators
- Strings, bytes, files, and I / O
- The main methods of working with strings, strings and Unicode
- Bytes Encodings Files and methods of working with files
- Classes
- Class declaration syntax
- Inheritance, method overloading, and super function
- Class decorators, @property decorator, magic methods, static methods and class methods
- Descriptors
- Constructor `_new_` and metaclasses
- Inheritance of built-in types
- Modules, Exceptions, and Context Managers
- Modules
- Import and from ... import statements
- Relative and absolute imports
- Exceptions, built-in exceptions, try statements - except - else-finally statements
- Context Managers and the contextlib module
- Iterators, generators
- Iterator protocol, operator-expression yield
- Generators, iterators, coroutines
- Built-in modules: functools, collections, itertools
- Detailed consideration of the listed modules

Examples of using Python

Testing properties, testing in the interpreter, doctoral

- Modules unittest, py.test, hypothesis
- Multithreading, coroutines, interprocess communication
- GIL
- Parallelism and competitiveness.
- Asyncio module
- Multiprocessing module
- Threading, queue, and concurrent.futures modules
- Bonus lesson (the topic will be selected based on the wishes of the audience)

1. Visit [Python Homework Help](#)
2. [Do My Homework](#)