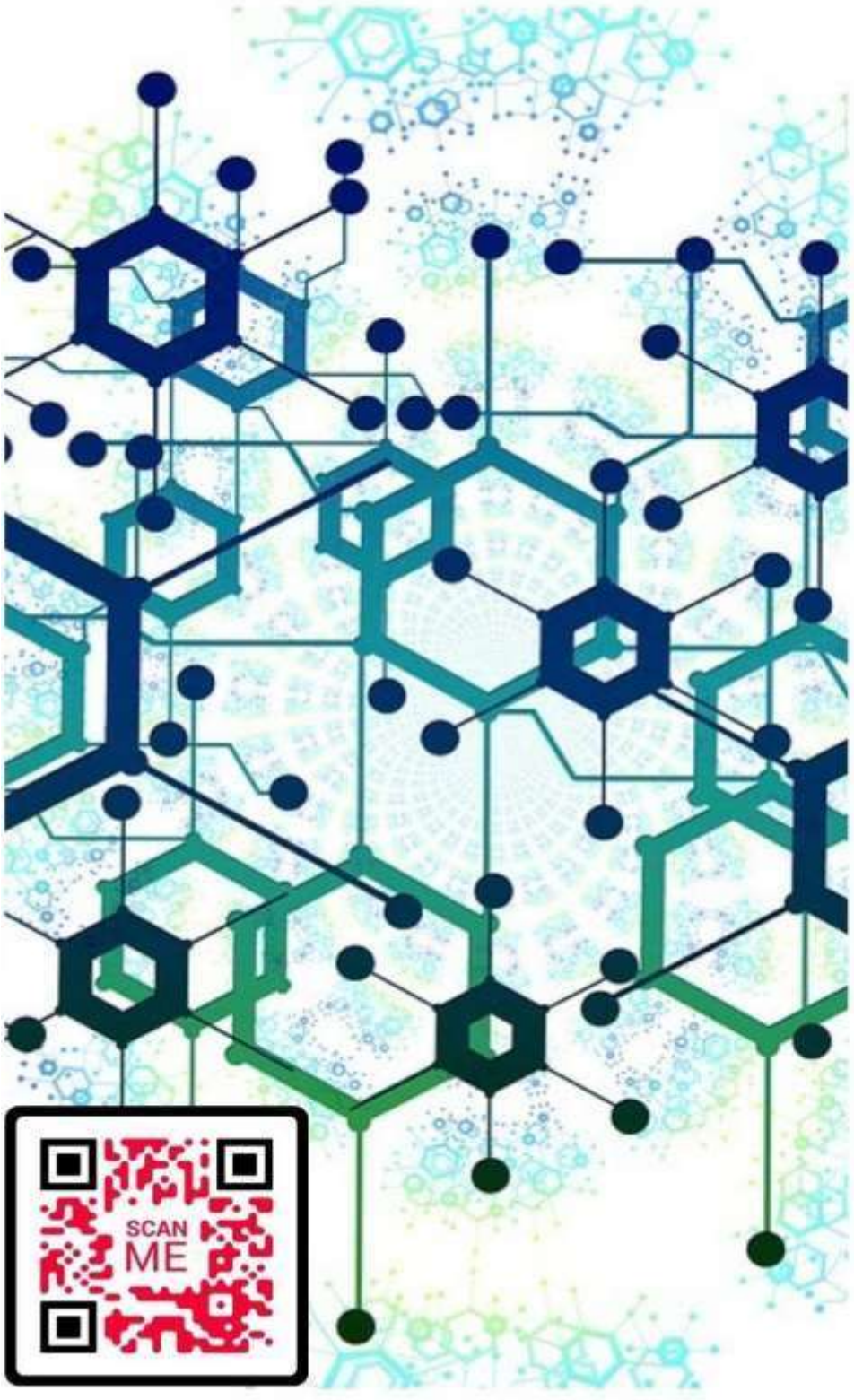
Indonesian Journal of Innovation Studies
Vol. 19 (2022): July 2022
Article type: (Innovation in Economics, Finance and Sustainable Development)

Table of Content

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 19 (2022): July 2022
Article type: (Innovation in Economics, Finance and Sustainable Development)



ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 19 (2022): July 2022
Article type: (Innovation in Economics, Finance and Sustainable Development)

Originality Statement

The author[s] declare that this article is their own work and to the best of their knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the published of any other published materials, except where due acknowledgement is made in the article. Any contribution made to the research by others, with whom author[s] have work, is explicitly acknowledged in the article.

Conflict of Interest Statement

The author[s] declare that this article was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright Statement

Indonesian Journal of Innovation Studies
Vol. 19 (2022): July 2022
Article type: (Innovation in Economics, Finance and Sustainable Development)

EDITORIAL TEAM

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 19 (2022): July 2022
Article type: (Innovation in Economics, Finance and Sustainable Development)

Article information

Check this article update (crossmark)

Check this article impact (*)

Save this article to Mendeley

(*) Time for indexing process is various, depends on indexing database platform

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 19 (2022): July 2022
Article type: (Innovation in Economics, Finance and Sustainable Development)

CLASSIFICATION OF PYTHON LANGUAGE MODULES

Author: Otahanov Nurillo Abdumalikovich
ID: nurillo_otaxanov
E-mail: nurillo_otaxanov@list.ru
Work: Namangan State University
Address: Namangan region, Namangan city, Uychi st. 316, index: 160119
Organisation: Namangan State University
Title: NamSU professori, PhD.
Phone: +998 (99) 101-08-40

Annotation: The Python programming language contains a large number of internal and external modules. In the process of teaching Python or developing a program in that language, it is important to identify the modules available and the areas to which they belong, in other words, to classify the modules into classes.

This article deals with the problem of classifying the internal and external modules of the Python programming language. In classifying, the author used a reliable source as well as personal experience and analysis. The article also provides information on the benefits of using DSLs.

Key words: Python, program, application, programming language, module, method, function, object, DSL, class, classification.

Introduction

Domain-specific programming language(DSPL), domain-specific language, DSL are programming languages with a high degree of abstraction, which is a tool for developing and solving programs related to a specific field. reflects the characteristics of a possible problem. Such programming languages use industry-specific terms and rules to collect, store, process, and transmit data.

In general, DSPL is not as complex as general-purpose programming languages such as Java, Pascal or C ++. Such programming languages are developed in collaboration with leading experts in the field of application and involve the use of modules.

The word module can be interpreted differently depending on the field of application. When it comes to programming languages, the concept of a module is interpreted as "a library of methods and functions, each of which is independent and designed to completely solve a small problem." As the word "library" suggests, each module can contain a large number of methods and functions. When it is necessary to use one or another component of the module, it is enough to refer to the appropriate method or function, specifying the parameters. Theoretically, the number of methods and functions in a module is not limited.

Generally, the use of programming languages, most of which are modular, does not require high skills, in other words, it is sufficient to have sufficient knowledge of the subject matter of the program and to have basic programming skills.

There are a number of advantages to using DSPL:

- much of the software development process has been simplified through the use of ready-made modules;
- saves programmers from writing less efficient operations, such as repeating the same small operations over and over again;

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

- facilitated by the possibility of developing a problem algorithm in the form of a logical project;
- some stages of program development can be effectively organized;
- programmers are freed from over-performing tasks;
- allows the use of accepted terms and rules in the field;
- establishes integrations between programmers and industry professionals;
- simplifies the process of editing (debugging) program code;
- facilitates reading, understanding and analysis of code;
- software improves product quality, reduces errors.

The idea of using the most appropriate programming language for the current problem, rather than a specific programming language, in the development of software led to the formation of the DSPL. Developers have developed custom DSLs for typical problems in their practice, and in return have expanded the capabilities of the programming language.

The term DSPL was first coined in 2004 by S. Dmitriev (founder and president of JetBrains) as a "machine-oriented programming language." Such approaches were developed in 2005 by Ch. Simoni's "mental programming" and M. Fowler's "Language tool: a new life of languages in the subject area."

Lisp was the first general-purpose programming language to include DSL programming capabilities.

Programming languages designed for a specific field and enriched with metad programming paradigms are an effective tool for software development and are widely used in practice today. Although such languages support a variety of programming paradigms, they are generally referred to as the most preferred paradigm.

Today, the Python programming language as DSPL is becoming increasingly popular. It should be noted that in 2021, the Python language became the number one programmer in the world in terms of demand and use. This is certainly not in vain. Because the Python programming language includes about 300 internal and about 100 external modules, it allows programmers to develop new applications in a simple, fast, inexpensive, cost-effective and timely manner. In this sense, it can also be interpreted as a modular programming language. However, since each Python module addresses a number of typical problems in a particular field and uses terms, concepts, and laws related to that area, it would be logical to consider it an object-oriented programming language (OOPL).

METHODS

In the process of developing applications in the Python programming language, it is necessary to find answers to a number of questions in order to take full advantage of its capabilities:

1. What Python modules are included?
2. What area does each module serve?
3. What methods and functions does each module include?

To answer the first two questions, you need to classify Pythom modules. The third question can be easily answered by programmers as they know the name of the field and module.

Our analysis of classifications in terms of subject-oriented programming has shown that the internal modules of the Python programming language cover typical issues in 24 areas. The classification was based on D. Hellman's book, The Standard Library of Python 3: A Guide to the Primeramas [2], pypi.org [3], kite.com [4], and the IDLE support system.

The ideas proposed by D. Hellman for the classification of standard modules of the Python language were taken as a basis. In his book, D. Hellman analyzes 117 modules, divides them into 19 classes, and describes the methods and functions that go into each class. These classes were enriched based on individual analysis of the remaining 200 modules and expanded to include 5 additional classes (Table 1.1).

RESULTS

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Classification of Python programming language modules. Table 1.1.

Class	List of modules
Working with texts	string, textwrap, re, difflib, grep, stringprep, extview;
Information structures	enum, collections, collection.abs, array, heapq, bisect, queue, struct, weakref, copy, pprint;
Algorithmic modules	functools, itertools, operator, contextlib;
Time and dates	time, datetime, calendar;
Mathematics	decimal, fractions, random, math, cmath, statistics, numeric;
File system	os.path, pathlib, glob, fnmath, linecache, tempfile, shutil, filecmp, mmap, mimetypes, codecs, io, code, filelist, genericpath, mimetypes, quopri, ntpath, outwin, plistlib, posixpath, setuptools;
Database, exchange and permanent storage	pickle, pickletools, copyreg, shelve, dbm, dbm.dumb, dbm.gnu, dbm.ndbm, sqlite3, xml.etree, Elementtree, csv, aifc, chunk, binascii, binhex, xdrlib, msilib, netrc, query, unicodedata, uu, spwd (unix), grp (Unix);
File archiving	zlib, gzip, bz2, tarfile, zipfile, lzma, zipapp, zipimport;
Cryptography	hashlib, hmac, crypt, encoding, h1, secrets;
Parallel computing, process, flow and satellite applications	ubprocess, signal, threading, multiprocessing, asyncio, concurrent.Futures;
Information exchange in the network	ipaddress, socket, selectors, select, socketserver, asyncore, asynchat, mb1, nntplib, poplib, prc, telnetlib;
Internet	urllib.parse, urllib.request, useurllib.Robotparser, base64, http.client, http.cookiejar, http.cookies, http.server, webrowser, browser, uuid, json, xml.prsclient, xml.prsserver, ftplib, html, html.entities, html.parser, h2, nturlpath, squeezer, ssl, xml.client, xml.server, smtplib, email, smtpd, imaplib, mailbox, mailcap;
Application development blocks	argparse, token, getopt, readline, getpass, cmd, wsgiref, shlex, pipes (unix), configparser, logging, pty (Unix), fileinput, fcntl (Unix), pyshell, Macosx, atexit, shed, py_compile, sidebar;
International and localization of applications	gettext, locale, zoneinfo;
Software development tools	pydoc, ParenMatch, sched, dataclasses, ctypes, curses, curses.ascii, curses.panel, curses.textpad, cProfile, hotshot, pstats, builtins, xxsubtype, doctest, unittest, trace, traceback, cgitb, pdb, debugger, timeit, profile, tabnanny, compileall, pyclbr, venv, ensurepip, bdb, contextvars, configdialog, help_about, idle, idle_test, idlelib, delegator, distutils, dynoption, e1, errno, faulthandler, formatter, h3, help, gyperparser, iomenu, keyword, lib2to3, marshal, pydoc, pyexpat, PyParsing, rlcompleter, search, sre_compile, sre_constants, sre_parse, stat, symtable, test, tty (UNIX), zoomheight, zzdummy;

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Program execution time tools	sys, msvcrt, site, codeop, os, nt, winreg, platform, resourse, gc, sysconfig, cgi, reprlib, runscript, tooltip, tracemalloc, types, typing, undo;
Language tools	warnings, abc, dis, inspect, tokenize;
Working with modules and packages	importlib, pkgutil, pip, zipimport, imp, modulefinder, multicall, opcode, pkg_resourse, redirector, runpy, syslog, nis (unix);
Working with images	antigravity, colorsys, colormath, config_key, graphlib, scrolledlist, tkinter, turtle, turtledemo;
Working with grammars	ast, optparse;
Working with sounds	audioop, autocomplete, sndhdr, sunau, wave, winsound;
Recognition of emblems	imghdr.

In Table 1.2 lists the standard (internal) modules for some classes.

Table of standard modules for individual classes. Table 1.2.

Modules	General content of module classes and methods
Working with texts	
string	class, method, function, and templates for working with texts
textwrap	text formatting methods
re	work with regular expressions
difflib	identify and process differences between sequences
grep	search for plain text files for clear lines.
stringprep	Line preparation in accordance with RFC 3453
textview	display and edit large amounts of formatted text.
Algorithmic modules	
functools	functions and decorators that allow for reusable programming and code
itertools	iterators and generators for use in functional programming
operator	a function-based interface for functional programming
contextlib	simplifies the resource management process
Mathematics	
decimal	working with real numbers in fixed accuracy
fractions	work with rational numbers
random	work with evenly distributed random numbers
math	work with various mathematical functions
cmath	Mathematical functions for complex numbers.
statistics	perform statistical calculations on different types of numbers
numeric	base digital abstract classes (complex, real, integer, etc.).

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

It should be noted that it takes a lot of physical effort and time to review the content of the modules included in the Python programming language, analyze them and describe the collected data in detail. It can be said that the methods and functions of each Python module require separate research. For example, the itertools module in the algorithmic class contains 18 methods and functions, each of which can solve typical problems of iterative algorithms. Python is a separate pedagogical issue in the process of teaching the programming language, as it is necessary to explain how to apply these methods and functions in practice.

It is also possible to enrich the Python language with external modules. Today, external modules have been developed in various fields and are widely used in programming practice. Personal research shows that there are about 100 such modules. Naturally, the subject-oriented programming approach to classifying these modules is also important for teaching them. Analysis of the external modules by industry showed that they could be divided into 13 classes. Table 1.3. lists these classes.

Classification of external modules of Python programming language.

Table 1.3.

Classes	List of modules
Working with graphical interfaces	wxPython, PyGObject, Pmw, WCK, Tix, tree;
Working with databases	MySQLdb, PyGreSQL, Gadfly, SQLAlchemy, KinterbasDB;
Working with web pages	Beautiful Soup, Flask, Scrape, Mechanize, Libgmail, Google, Maps, Requests, Selenium, pyquery, WhatWaf, Twisted, jabberpy, Sentry;
Images and videos	PIL, Gdmodule, Video Capture, pyscreeshot, MoviePy, VPython;
Data Science and Mathematics	SciPy (Say pay), Matplotlib, Pandas, NumPy, Sympy, Teano, Chainer;
Game development	Pygame, Pyglet, pyOpenGl;
Working with sounds	pySonic, pyMedia, PMIDI, Mutagen;
For Mac OS	py2app, PyObjC;
For USB and serial ports	PyUSB, PySerial, USPP;
Neural networks	Keras, PyTorch, sikit-learn, Som-tsp, Pylearn2, Nilearn, Detectron;
For the software development process	pyExpect, Pipenv, Multidiff, Cookiecutter, Python-fire;
cryptocurrency	SimpleCoin;
work with files	Gensim.

Table 1.4. provides information on the modules belonging to some of these classes.

External modules for some subject areas. Table 1.4.

Working with graphical interfaces	
wxPython	Creating real user interfaces on operating systems such as Windows, Mac and Linux, Unix.
PyGObject	Connect to libraries based on GTK, GStreamer, WebKitGTK, GLib, GIO, Gobject and others.
Pmw	A set of tools for creating high-level composite widgets in Python (using Tkinter).
WCK	An API extension that allows you to work with all types of custom widgets in Python.

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Tix	Powerful user interface that can be used to expand the capabilities of Tcl / Tk and Python applications
Ma'lumotlar bazasi bilan ishlash	
MySQLdb	Python DB API-2.0 compatible interface for accessing MySQL databases.
PyGreSQL	Interface for PostgreSQL database
Gadfly .	A simple relational database implemented in Python (based on SQL).
SQLAlchemy	Database module used by ORM.
KinterbasDB	FireBird relational database module.
Working with web pages	
Beautiful Soup	Module for fast parsing of HTML and XML files.
Flask	web application development
Scrape	module for web browser and skripping.
Mechanize	browser emulator for interaction with web browsers.
Libgmail	Gmail allows you to work with the google system.
Google Maps	Web services for the Google Maps platform.
Requests	Makes it easier to work with HTTP / 1/1 queries.
Selenium	makes it easy to open web pages, fill in fields, click buttons, and submit forms.
pyquery	Create jQuery queries in XML documents.
WhatWaf	bypass the web application protection system
Working with images and videos	
PIL	(Python Imaging library) image processing.
Gdmodule	Interface for GD library.
Video Capture	work with webcam, TV card and other video devices.
pyscreeshot	non-stop screenshots without installing external libraries.
MoviePy	video editing module.
Data Science and Mathematical Computations	
ScyPy (Say pay)	Ecosystem for mathematics, science and engineering.
Matplotlib	Module for creating 2D graphics.
Pandas	a module for data analysis, capture and processing.
NumPy	module for working with multidimensional arrays.
Teano	identify, optimize, and effectively evaluate mathematical expressions involving multidimensional arrays.
Game development	
Pygame	Video game module.
Pyglet	Multiple visual effects module for games.
pyOpenGl	Module for working with OpenGl.

DISCUSSION

Tables 1.1 - 1.4 show that the Python programming language allows you to accelerate the process of developing a program for a large number of typical problems in many subject areas. It should be noted that the modules of the Python programming language can be expanded by removing the "obsolete" ones depending on the new versions of the language and adding new ones, and, accordingly, the classes covered by Python can be expanded.

The most important advantage of DSPL is that it saves costs (memory, time, money, human and other resources) associated with software development. In addition, DSPL allows programmers to significantly reduce the complexity of issues. This suggests that even those with a superficial knowledge and skills in Python programming can easily develop a program. This

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

means that until recently, there have been opportunities for other industry professionals to develop software products that can only be created by skilled programmers.

To take advantage of such opportunities, it is necessary to reconsider and reorganize the content of teaching computer science in higher and secondary education, which is designed to train specialists in the field, which requires a wide range of software.

LIST OF REFERENCES

1. https://www.it-world.ru/tech/choice/175385.html.
2. Хеллман Д. Стандартная библиотека Python 3: справочник с примерами. — СПб. : “Диалектика”, 2019. — 1376 с.
3. https://pypi.org/project/
4. https://www.kite.com/python/docs/

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo