

České vysoké učení technické v Praze FIT

Programování v Pythonu

Jiří Znamenáček

Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.

Praha & EU: Investujeme do vaší budoucnosti



Python - Některé zajímavé vestavěné moduly

Úvod

Už od historické doby se o Python'u říká, že přichází s „batteries included“. Důvodem je jeho *velmi obsáhlá* vestavěná knihovna dostupná prakticky v každé instalaci Python'u. Budete-li se pouštět do nějakého většího pythoního projektu, rozhodně se vyplatí do knihovny alespoň nahlédnout – často asi zjistíte, že obsahuje něco, co vám přinejmenším zjednoduší práci (pokud už nebude obsahovat dokonce plnou implementaci toho, co zrovna hledáte).

→ Některé z modulů jsou dostupné pouze na některých platformách případně pouze v rámci některých instalací. Většinou se však jedná o platformně specifické záležitosti (např. modul `curses` pro pokročilou obsluhu terminálu existuje pouze pro unixové platformy) a dokumentace se o rozdílech vždy zmiňuje.

Přímo z příkazové řádky můžete zjistit vám aktuálně dostupné moduly pomocí příkazu `help('modules')`. Výstup vás nejspíš ohromí:

<code>__future__</code>	<code>code</code>	<code>ntsecuritycon</code>	<code>this</code>
<code>_abcoll</code>	<code>codecs</code>	<code>nturl2path</code>	<code>thread</code>
<code>_ast</code>	<code>codeop</code>	<code>numbers</code>	<code>time</code>
<code>_bisect</code>	<code>collections</code>	<code>odbc</code>	<code>timeit</code>
<code>_codecs</code>	<code>colorsys</code>	<code>opcode</code>	<code>timer</code>
<code>_codecs_cn</code>	<code>commctrl</code>	<code>operator</code>	<code>tkinte</code>
<code>_codecs_hk</code>	<code>compileall</code>	<code>optparse</code>	<code>token</code>
<code>_codecs_iso2022</code>	<code>concurrent</code>	<code>os</code>	<code>tokeni</code>
<code>_codecs_jp</code>	<code>configparser</code>	<code>os2emxpath</code>	<code>trace</code>
<code>_codecs_kr</code>	<code>contextlib</code>	<code>parser</code>	<code>traceb</code>
<code>_codecs_tw</code>	<code>copy</code>	<code>pdb</code>	<code>tty</code>
<code>_collections</code>	<code>copyreg</code>	<code>perfmon</code>	<code>turtle</code>
<code>_compat_pickle</code>	<code>csv</code>	<code>pickle</code>	<code>turtle</code>
<code>_csv</code>	<code>ctypes</code>	<code>pickletools</code>	<code>types</code>
<code>_ctypes</code>	<code>curses</code>	<code>pipes</code>	<code>unicod</code>
<code>_ctypes_test</code>	<code>datetime</code>	<code>pkg_resources</code>	<code>unitte</code>
<code>_datetime</code>	<code>dbi</code>	<code>pkgutil</code>	<code>urllib</code>
<code>_dummy_thread</code>	<code>dbm</code>	<code>platform</code>	<code>uu</code>
<code>_elementtree</code>	<code>dde</code>	<code>plistlib</code>	<code>uuid</code>
<code>_functools</code>	<code>decimal</code>	<code>poplib</code>	<code>warnin</code>
<code>_hashlib</code>	<code>difflib</code>	<code>posixpath</code>	<code>wave</code>
<code>_heapq</code>	<code>dis</code>	<code>pprint</code>	<code>weakre</code>
<code>_io</code>	<code>distutils</code>	<code>profile</code>	<code>webbro</code>
<code>_json</code>	<code>doctest</code>	<code>pstats</code>	<code>win2kr</code>
<code>_locale</code>	<code>dummy_threading</code>	<code>pty</code>	<code>win32a</code>
<code>_lsprof</code>	<code>easy_install</code>	<code>py_compile</code>	<code>win32c</code>
<code>_markupbase</code>	<code>email</code>	<code>pyclbr</code>	<code>win32c</code>
<code>_md5</code>	<code>encodings</code>	<code>pydoc</code>	<code>win32c</code>
<code>_msi</code>	<code>errno</code>	<code>pydoc_data</code>	<code>win32c</code>
<code>_multibytecodec</code>	<code>filecmp</code>	<code>pyexpat</code>	<code>win32c</code>
<code>_multiprocessing</code>	<code>fileinput</code>	<code>pypm</code>	<code>win32c</code>
<code>_pickle</code>	<code>fnmatch</code>	<code>pythoncom</code>	<code>win32c</code>
<code>_pyio</code>	<code>formatter</code>	<code>pywin</code>	<code>win32e</code>
<code>_random</code>	<code>fractions</code>	<code>pywin32_testutil</code>	<code>win32e</code>
<code>_sha1</code>	<code>ftplib</code>	<code>pywintypes</code>	<code>win32e</code>
<code>_sha256</code>	<code>functools</code>	<code>queue</code>	<code>win32f</code>
<code>_sha512</code>	<code>gc</code>	<code>quopri</code>	<code>win32g</code>
<code>_socket</code>	<code>genericpath</code>	<code>random</code>	<code>win32g</code>
<code>_sqlite3</code>	<code>getopt</code>	<code>rasutil</code>	<code>win32h</code>
<code>_sre</code>	<code>getpass</code>	<code>re</code>	<code>win32i</code>
<code>_ssl</code>	<code>gettext</code>	<code>regcheck</code>	<code>win32i</code>
<code>_string</code>	<code>glob</code>	<code>regutil</code>	<code>win32j</code>
<code>_strptime</code>	<code>gzip</code>	<code>reprlib</code>	<code>win32l</code>
<code>_struct</code>	<code>hashlib</code>	<code>rlcompleter</code>	<code>win32n</code>
<code>_subprocess</code>	<code>heapq</code>	<code>runpy</code>	<code>win32n</code>
<code>_symtable</code>	<code>hmac</code>	<code>sched</code>	<code>win32p</code>
<code>_testcapi</code>	<code>html</code>	<code>select</code>	<code>win32p</code>
<code>_thread</code>	<code>http</code>	<code>servicemanager</code>	<code>win32p</code>
<code>_threading_local</code>	<code>idlelib</code>	<code>setuptools</code>	<code>win32p</code>
<code>_tkinter</code>	<code>imaplib</code>	<code>shelve</code>	<code>win32p</code>
<code>_warnings</code>	<code>imghdr</code>	<code>shlex</code>	<code>win32p</code>
<code>_weakref</code>	<code>imp</code>	<code>shutil</code>	<code>win32p</code>
<code>_weakrefset</code>	<code>importlib</code>	<code>signal</code>	<code>win32r</code>
<code>win32sysloader</code>	<code>inspect</code>	<code>site</code>	<code>win32r</code>

- Instalace Python'u *ActiveState* pro platformu Windows zjevně obsahuje enormní množství platformě specifických modulů `win32` NĚCO.

Poznámky

I. Moduly prošly do verze 3.x dosti důkladnou revizí (některé části byly popřesouvány jinam, sem tam něco přejmenováno a podobně).

II. Globální vestavěné funkce podobná revize bohužel skoro minula a jeví tak jasné známky historického vývoje.

- Většina vestavěných fukcí jsou vlastně konstruktory různých typů, případně obecné funkce pro práci s nimi. Zatoulalo se sem však i pár fukcí, které by měly být jinde – například `abs()` nebo `pow()`. Podobně tu zbylo pár funkcí pro funkcionální programování (`map()`, `filter()`), které byly postupně nahrazeny jinými konstrukcemi (*comprehension*).

string

Při práci budete naprosto běžně používat datový typ *řetězec* s jeho běžnou armádou atributů a metod. Ale vyplatí se vědět, že máte k dispozici o modul *string*, v němž jsou mimo jiné definovány následující velmi užitečné konstanty:

- `string.ascii_lowercase` – nezávisle na nastavení *locale* vždy následující řetězec: 'abcdefghijklmnopqrstuvwxyz'.
- `string.ascii_uppercase` – nezávisle na nastavení *locale* vždy následující řetězec: 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.
- `string.ascii_letters` – spojení *ascii_lowercase* a *ascii_uppercase*
- `string.digits` – řetězec '0123456789'
- `string.hexdigits` – řetězec '0123456789abcdefABCDEF'
- `string.octdigits` – řetězec '01234567'
- `string.punctuation` – řetězec ASCII-znaků, které odpovídající céčkovský *locale* pokládá za interpunkci
- `string.whitespace` – řetězec ASCII-znaků, které jsou pokládány za

whitespaces, čili mimo jiné mezera, tabulátor, odřádkování atd.

- `string.printable` – řetězec ASCII-znaků, které jsou pokládány za tisknutelné; prakticky kombinace *digits*, *ascii_letters*, *punctuation* a *whitespace*

Kromě výše uvedených konstant zde dále najdete prostředky pro formátování řetězců a jednoduché šablonování.

builtins & keywords & types

I. Modul *builtins* drží seznam všech v Python'u dostupných identifikátorů:

```
>>> import builtins
>>> dir(builtins)

['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BufferError', 'BytesWarning', 'DeprecationWarning', 'EOFError', 'EnvironmentError', 'Exception', 'False', 'FloatingPointError', 'FutureWarning', 'GeneratorExit', 'IOError', 'ImportError', 'ImportWarning', 'IndentationError', 'IndexError', 'KeyError', 'KeyboardInterrupt', 'LookupError', 'MemoryError', 'NameError', 'None', 'NotImplemented', 'NotImplementedError', 'OSError', 'OverflowError', 'PendingDeprecationWarning', 'ReferenceError', 'RuntimeWarning', 'StopIteration', 'SyntaxError', 'SyntaxWarning', 'SystemExit', 'TabError', 'True', 'TypeError', 'UnboundLocalError', 'UnicodeDecodeError', 'UnicodeEncodeError', 'UnicodeError', 'UnicodeWarning', 'UserWarning', 'ValueError', 'Warning', 'ZeroDivisionError', '__build_class__', '__debug__', '__doc__', '__import__', '__name__', '__package__', '__spec__', 'abs', 'all', 'any', 'ascii', 'bin', 'bool', 'bytearray', 'bytes', 'classmethod', 'compile', 'complex', 'copyright', 'credits', 'delattr', 'dir', 'divmod', 'enumerate', 'eval', 'exec', 'exit', 'filter', 'float', 'frozenset', 'getattr', 'globals', 'hasattr', 'hash', 'help', 'hex', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len', 'license', 'locals', 'map', 'max', 'memoryview', 'min', 'next', 'object', 'oct', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr', 'reversed', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str', 'sum', 'tuple', 'type', 'vars', 'zip']
```

- Jinými slovy řečeno jsou tyto identifikátory dostupné též pod tečkovou notací, tj. např. *builtins.open()* apod.
- Ve spoustě objektů se k němu můžeme dostat přes jejich atribut `__builtins__`.

II. Na podobnou notu modul *keyword* drží seznam klíčových slov v Python'u:

```
>>> import keyword

>>> dir(keyword)
['__all__', '__builtins__', '__cached__', '__doc__', '__file__', '
__package__', 'iskeyword', 'kwlist', 'main']

>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class',
'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from',
'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'p
'return', 'try', 'while', 'with', 'yield']
```

Metoda *iskeyword(s)* se dá použít pro zjištění, zda daný řetězec představuje klíčové slovo či nikoli.

III. A do třetice modul *types* drží jména některých pythoních typů:

```
>>> import types

>>> dir(types)
['BuiltinFunctionType', 'BuiltinMethodType', 'CodeType', 'FrameType',
'FunctionType', 'GeneratorType', 'GetSetDescriptorType', 'LambdaType',
'MemberDescriptorType', 'MethodType', 'ModuleType', 'TracebackType',
'__builtins__', '__cached__', '__doc__', '__file__', '__name__',
```

Typicky vhodné pro použití v rámci dotazů *isinstance()* nebo *issubclass()*.

Všehochuť na začátek

- pprint – *pretty-printer* pro nejrůznější datové struktury
- re – regulární výrazy
- difflib – zjišťování rozdílů mezi sekvencemi

Další datové typy

- datetime – operace s datem a časem
- heapq – implementace *heap queue* algoritmu
- array – znaková a numerická pole

Čísla a matematika

- `math` – matematické funkce
- `decimal` – numerické operace v přesném desítkovém vyjádření
- `fractions` – práce se zlomky
- `random` – generátor (pseudo)náhodných čísel

Souborový systém

- `os.path` – manipulace s cestou
- `shutil` – vysokoúrovňové operace
- `filecmp` – porovnávání souborů a adresářů
- `tempfile` – práce s dočasnými objekty (napříč různými OS)
- `linecache` – náhodný přístup k řádkám souboru

Vrstva ukládání dat

- `pickle` – (de)serializace pythoních datových struktur
- `sqlite3` – DB-API 2.0 interface pro SQLite

Komprimace a archivace

- `zlib`, `gzip` – podpora *gzip* (*zlib* mimo jiné poskytuje implementaci algoritmu CRC32)
- `bz2` – podpora pro *bz2*
- `zipfile` – podpora *zip*
- `tarfile` – podpora *tar*

Formáty souborů

- `csv` – podpora pro CSV
- `configparser` – parser pro jednoduché konfigurační soubory (slušně konfigurovatelný, zvláště od verze 3.2)
- `netrc` – *netrc*-soubory ftp-programů
- `xdrlib` – podpora pro *External Data Representation Standard (RFC 1014)*
- `plistlib` – podpora pro *.plist*-soubory v Mac OS X

Kryptografie

- `hashlib` – podpora pro SHA1, SHA224, SHA256, SHA384, SHA512 a MD5
- `hmac` – podpora pro HMAC-algoritmus (RFC 2104)

Operační systém

- `os` – nejrozumnější interfejsy pro interakci s OS
- `time` – operace s časem
- `timeit` – modul pro měření doby vykonávání kódu
- `optparse` – parser parametrů příkazové řádky
- `argparse` – nejnovější verze parseru parametrů příkazové řádky
- `curses` – „grafická“ obsluha terminálu
- `logging` – logovací modul
- `platform` – informace o platformě
- `threading` – vysokoúrovňový interfejs pro vlákna

Komunikace mezi procesy

PS: A to jak na stejném stroji, tak i po síti.

- `subprocess` – obsluha podprocesů
- `socket` – nízkoúrovňový přístup k síťovým službám
- `ssl` – SSL-wrapper pro socket-objekty

Strukturovaná textová data

- `html` – podpora HTML a XHTML
- `xml` – podpora XML (a to jak DOM, tak i SAX; obsahuje i zjednodušený přístup k DOMu pomocí modulu *etree*)

PS: Populárním externím nástrojem na zpracování HTML/XML je knihovna [Beautiful Soup](#), která si dobře poradí s HTML, které dokáže být hodně „zamotané“.

Internet - data

- `email`, `mailbox` – obsluha mailboxu a imejlů
- `mimetypes` – rozpoznávání typu souborů
- `base64`, `json`, `uu` – přenos dat
- `binhex`, `binascii` – konverze

Internet - protokoly

- `urllib` – přístup na URL
- `uuid` – UUID-objekty podle RFC 4122
- `http`, `ftplib`, `telnetlib` – protokoly HTTP, FTP a telnet
- `cgi`, `wsgi`, `xmlrpc` – podpora CGI, WSGI a XML-RPC
- `poplib`, `imaplib`, `nntplib`, `smtplib` – podpora POP3, IMAP4, NNTP a SMTP

Multimédia

- `aifc`, `sunau`, `wave` – podpora AIFF, AIFC, AU, WAV
- `colorsys` – konverze mezi prostory barev
- `imghdr`, `sndhdr` – detekce typu obrázku a zvuku

Internacionalizace

Aneb moduly pro práci v mezinárodním jazykovém prostředí:

- `gettext` – modul poskytující služby pro vícejazykové prostředí (*i18n* a *l10n*)
- `locale` – přístup na databázi a služby obsluhy národního prostředí ve standardu POSIX

Pár drobtů na konec

- `sys` – prostředí interpretru
- `cmd` – příkazová řádka
- `shlex` – jednoduchá lexikální analýza
- `ctypes` – *foreign function library* (pro použití zvláště jako wrapper nad externími knihovnami)
- `struct` – konverze mezi céčkovskými *struct* (reprezentovány jako pythoní *byte*-objekty) a pythoními datovými strukturami
- `doctest`, `unittest`, `2to3`, `traceback` – vývoj
- `pdb`, `timeit`, `trace` – debugování
- - .