Python

Vestavěné funkce a vybrané knihovny

VŠCHT

2019

Vestavěné funkce

Vestavěné funkce přehled

Vestavěné funkce - seznam				
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

Vybrané vestavěné funkce

Popis vestavěných funkcí

- *abs*(*arg*) - funkce vrátí absolutní hodnotu argumentu (integeru, floatu). Pokud je argumentem komplexní číslo, vrací také jeho absolutní hodnotu
- *all*(*arg*) - testuje, zda jsou všechny elementy iterovatelného argumentu True, pokud ano, vrátí True
- *any*(*arg*) - testuje, zda je nějaký element argumentu True, pokud ano, vrací True
- *ascii*(*arg*) - vrací tisknutelný řetezec, který reprezentuje argument
- *bin*(*arg*) - převede číslo typu integer do binárního stringu s prefixem 0*b*
- *bool*(*arg*) - pokud je argument False nebo chybí, vrací False, jinak převede argument na True

Vestavěné funkce - příklad

```
print(abs(-1.678))  # prints 1.678
print(all([True, False]))  # prints False
print(all([True, True, True]))  # prints True
print(all([False]))  # prints False
print(any([True, False]))  # prints True
print(any([True, True, True]))  # prints True
print(any([False, False]))  # prints False
print(ascii('a xy'))  # prints 'a\xddxy'
print(bin(-10))  # prints -0b1010
print(bool(), bool('something'))  # prints False True
```

Vybrané vestavěné funkce - pokračování

Popis vestavěných funkcí

- *bytearray*(*source*) - vytvoří nové pole bytů, source specifikuje pole
- *bytes*(*source*) - vytvoří nový objekt typu byte
- *chr*(*i*) - vrátí řetězec který obsahuje jeden znak jehož unicode kód je argumentem funkce
- *complex*([*real*, *imag*]) - vytvoří komplexní číslo se zadanou reálnou a imaginární částí
- *float*(*x*) - vrátí číslo typu float vytvořené ze zadaného řetězce
- *hash*(*arg*) - vrátí hash zadaného argumentu
- *hex*(*x*) - převede číslo do hexadecimálního řetězce s prefixem "0x"
- *input*() - funkce čte vstup (klávesnice) a po stisknutí Enter ho převede do řetězce
- *map*(*function*, *iterable*) - použije funkci function na všechny prvky proměnné iterable a vrátí příslušný iterátor

Vestavěné funkce - pokračování 1. - příklad

```
some_string = "string"
print(bytearray(3))  # prints bytearray(b'\x00\x00\x00')
print(bytearray([1, 2, 3]))  # prints bytearray(b'\x01\x02\x03')
print(bytearray(some_string, 'utf-8'))  # prints bytearray(b'string')
print(bytes(3))  # prints b'\x00\x00\x00'
print(bytes([1, 2, 3]))  # prints b'\x01\x02\x03'
print(bytes(some_string, 'utf-8'))  # prints b'string'
print(chr(64), chr(70))  # prints @ F
print(complex(1, -5))  # prints (1-5j)
print(hash(some_string))  # prints hash (always different)
print(hex(33))  # prints 0x21
print(list(map(list, some_string)))  # prints [['s'], ['t'], ['r'], ['i'], ['n'], ['g']]
print(input())  # prints users input after pressing Enter
```

Vybrané vestavěné funkce - pokračování 2.

Popis vestavěných funkcí

- *oct*(*x*) - převede číslo do řetezce reprezentujícího oktalovou hodnotu, řetězec má prefix "0o"
- *ord*(*c*) - vrací Unicode kód zadaného znaku
- *repr*(*object*) - vrací řetězec, který obsahuje tisknutelnou reprezentaci objektu
- *round*(*number*) - zaokrouhlí číslo
- *reversed*(*iterable*) - vrátí iterátor v obráceném pořadí
- *sum*(*iterable*) - sečte všechny prvky proměnné iterable
- *zip*(∗*iterables*) - vrací iterátor n-tic vytvořených spojením elementů jednotlivých iterátorů na vstupu

Vestavěné funkce - pokračování 2. - příklad

```
obj = "some string"
print(oct(19))  # prints 0o23
print(ord("a"))  # prints 97
print(repr(obj))  # prints 'some string'
print(round(7.7598, ndigits=1))  # prints 7.8
print(round(7.7598))  # prints 8
print(reversed([1, 2, 3]))
# prints <list_reverseiterator object at 0x0083B3D0>
print(list(reversed([1, 2, 3])))  # prints [3, 2, 1]
print(sum([1, 2, 3]))  # prints 6
print(zip([1, 2], [3, 4]))
#<zip object at 0x00B473C8>
print(list(zip([1, 2], [3, 4]))) # prints [(1, 3), (2, 4)]
```

Vybrané knihovny

Přehled populárních knihoven

- **numpy** - knihovna pro práci s poli, matematické funkce (bude probráno)
- **scipy** - knihovna pro inženýry, obsahuje optimalizační funkce, interpolační funkce, statistické funkce, funkce pro zpracování signálů a další (bude probráno)
- **pandas** - knihovna pro práci s Big Data
- **matplotlib** - knihovna pro tvorbu grafů, obrázků, atd. (bude probráno)
- **scrapy** - knihovna pro síťovou komunikaci a práci s pakety
- **PyQt** - knihovna pro tvorbu GUI
- **OpenCV** - knihovna pro zpracování obrazů
- **Django** - framework pro tvorbu webových stránek
- **scikit-learn** - knihovna pro machine-learning

Vybrané knihovny - pokračování

Přehled populárních knihoven

- **pygame** - knihovna pro tvorbu her
- **requests** - knihovna pro práci s http
- **Beautifulsoup** - HTML a XML parser
- **SymPy** - knihovna pro symbolické výpočty
- **Padasip** - knihovna adaptivní zpracování signálů
- **TensorFlow** - knihovna vyvinutá Google pro deep learning

Práce s modulem resp. knihovnou

Každý modul je nutné před jeho použitím importovat. Podle konvence se import provádí vždy na začátku souboru s kódem, ve kterém je modul použit.

Numpy - úvod

Numpy

Numpy umožňuje jednoduchou práci s poli a používání řady matematických funkcí.

- vytvoření pole - $array(shape)$
- konverze seznamu do pole - $asarray(x)$
- uložení pole do souboru - $savetxt(x)$
- vytvoření pole ze souboru - $fromfile(file, separator)$
- nahrání dat z textového souboru do pole - $loadtxt(fname)$
- vektor rovnoměrně rozložených hodnot - $arange(start, stop, step)$
- vektor rovnoměrně rozložených hodnot včetně obou krajních mezí - $linspace(start, stop, number)$
- matice souřadnic vytvořená z vektorů souřadnic - $meshgrid(x, y)$

Numpy - základy - příklad

```
import numpy as np

x = np.array((1, 2, 3))
print(np.array([1, 2, 3]))
print(np.asarray([1, 2, 3]))
print(np.arange(0, 1, 0.2))
print(np.linspace(0, 1, 6))
x_val, y_val = np.meshgrid([1, 2], [3, 4])
print(x_val, y_val)
np.savetxt('data.txt', x)
print(np.fromfile('data.txt', sep='\n'))
print(np.loadtxt('data.txt'))
```