Python. Wprowadzenie. Wydanie IV

Mark Lutz

Drogi Czytelniku! Poniżej zamieszczona jest errata do książki:
"Python. Wprowadzenie. Wydanie IV"

Jest to lista błędów znalezionych po opublikowaniu książki, zgłoszonych i zaakceptowanych przez naszą redakcję. Pragniemy, aby nasze publikacje były wiarygodne i spełniały Twoje oczekiwania. Zapoznaj się z poniższą listą. Jeśli masz dodatkowe zastrzeżenia, możesz je zgłosić pod adresem https://helion.pl/user/erraty

Strona	Linia	Jest	Powinno
160	18	liczę	liczbę
163	cała strona		Strona 163
164	cała strona		Strona 164
169	3 kolejne linie kodu od góry	math.floor(2.567),math.floor(-2.567) (2,3) math.trunc(2.567,math.trunc(2.567) (2,2) int(2.567),int(2.567) (2,2)	math.floor(2.567),math.floor(-2.567) (2,-3) math.trunc(2.567,math.trunc(-2.567) (2,-2) int(2.567),int(-2.567) (2,-2)
169	cała strona		Strona 169
213		>>> while B: != ″:	>>> while B != ″:
217		S = S[:where] + ′MIELONKA′ + S[(where+4):]	S = S[:where] + ′MIELONKA′ + S[(where+5):]

232	cała strona		Strona 232
243	cała strona		Strona 243
373	cała strona		Strona 373
466	9	Ponieważ jednak nie można zmodyfikować obiektów zmiennych w miejscu, efekt jest taki, jakbyśmy sporządzili kopię.	Ponieważ jednak nie można zmodyfikować obiektów niezmiennych w miejscu, efekt jest taki, jakbyśmy sporządzili kopię.
527	cała strona		Strona 527
528	cała strona		Strona 528
529	cała strona		Strona 529
612	cała strona		Strona 612
730	cała strona		Strona 730
731	cała strona		Strona 731
739	cała strona		Strona 739

Poniżej znajduje się lista błędów znalezionych przez czytelników, ale jeszcze nie potwierdzonych przez Redakcję:

Strona	Linia	Jest	Powinno

956	**33**	**Metody__getattr__ oraz__setattr__ służące do przekierowywania niezdefiniowanych pobrań atrybutów, a także wszystkich przypisań atrybutów do ogólnych metod programów obsługi.**	**Metody __getattr__ oraz __setattr__ służące do przekierowywania pobrań niezdefiniowanych atrybutów, a także wszystkich przypisań atrybutów do ogólnych metod programów obsługi.**
960	**15**	**Po wykonaniu do udekorowanej metody automatycznie przekazywany jest pierwszy argument funkcji wbudowanej property.**	**Po wykonaniu dekorowana metoda automatycznie przekazywana jest jako pierwszy argument funkcji wbudowanej property.**
970	**36**	**By przechwytywać również zmianę atrybutu przez jego przypisanie, musimy utworzyć kod __setattr__ - metody przeciążania atrybutów wykonywanej dla każdego ich pobrania,**	**By przechwytywać również zmianę atrybutu przez jego przypisanie, musimy utworzyć kod __setattr__ - metody przeciążania atrybutów wykonywanej dla każdego ich przypisania,**

971	13,14, 15, 16	def __getattr__(self, nazwa):
		def __getattribute__(self, nazwa):
		def __setattr__(self, nazwa):
		def __delattr__(self, nazwa):
			def __getattr__(self, name):
			def __getattribute__(self, name):
			def __setattr__(self, name):
			def __delattr__(self, name):