

Perl dla każdego

Laura Lemay

Drogi Czytelniku! Poniżej zamieszczona jest errata do książki:
"Perl dla każdego"

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://ebookpoint.pl/user/erraty

Strona	Linia	Jest	Powinno
65	wiersz 6 od góry	Lt	lt
74	wiersz 18 od góry	$fahr = $cel * 5 / 9 +32	$fahr = $cel * 9 / 5 +32
91	wiersz 13 od góry	znich	z nich
91	wiersz 6 od góry	prefiksowym operatorem inkrementacji ($x++) i postfiksowym operatorem inkrementacji (++$x)	prefiksowym operatorem inkrementacji (++$x) i postfiksowym operatorem inkrementacji ($x++)
93	wiersz 2 od dołu	$spcaje	$spacje
97	wiersz 1 od dołu	@nums = (@nums , 5);	@nums = (@liczby , 5);

97	wiersz 15 od góry	liczby@	@liczby
97	wiersz 4 od dołu	@poloczenie = (@liczby, @lancuchy);	@polaczenie = (@liczby, @mieszanka);
97	wiersz 9 od góry	przecinakami	przcinkami
104	wiersz 14 od dołu	chomp ($input = <STDIN>);	chomp ($dane = <STDIN>);
137	wiersz 13 od dołu	$freq{$dane}++;	$freq{$liczba}++;
137	wiersz 17 od dołu	chomp ($input = <STDIN>);	chomp ($dane = <STDIN>);
145	wiersz 6 od dołu	$input	$dane
159	wiersz 3 od góry	while (<>)) {	while (<>) {
166	wiersz 24 od góry	W kodzie występuje tylko jeden błąd: słowo elseif nie jest słowem kluczowym Perla, zamiast niego należy uzyć elsif.	W kodzie występują tylko dwa błędy: słowo elseif nie jest słowem kluczowym Perla, zamiast niego należy uzyć elsif. Nie ma także słowa kluczowego then, należy je opuścić.
185	wiersz 10 od dołu	elsif ($a gt $b { $wynik =1; }	elsif ($a gt $b) { $wynik =1; }

194	wiersz 13 od góry	push @listy @lista1;	push @listy, @lista1;
196	wiersz 11 od dołu	@listaliczba	@listaliczb
196	wiersz 14 od góry	(0, 1, 2, 3, 4, 5, 5.1, 5.2, 5.3, 6, 7, 8, 9)	(0, 1, 2, 3, 4, 5.1, 5.2, 5.3, 5, 6, 7, 8, 9)
196	wiersz 8 od dołu	shift @listaliczb;	#shift @listaliczb;
196	wiersz 9 od dołu	@lista1 = splice(@listaliczb, 1, $listaliczb[0]);	@lista1 = splice(@listaliczb, 0, $listaliczb[0]);
197	wiersz 5 od dołu	(@liczba ** 2)	($liczba ** 2)
199	wiersz 2 od dołu	push @wynik, $l;	push @wynik, $l; }
199	wiersz 21 od dołu	tablicy @listy	tablicy @lista
206	wiersz 6 od dołu	@znaki = split (' ', $str)	@znaki = split //, $str
212	wiersz 13 od góry	/^koniec/	/koniec$/
216	wiersz 13 od góry	np. [/0-9]/, [/a-z/]	np. /[0-9]/, /[a-z]/

216	wiersz 4 od dołu	to /^12/ nie znaczy wcale	to /[^12]/ nie znaczy wcale
216	wiersz 5 od dołu	Jezeli np. wzorzec /12/	Jezeli np. wzorzec /[12]/
221	wiersz 17 od dołu	W konca	W końcu
226	wiersz 1 od dołu	maksimum dwóch liczb po prawej stronie	dokładnie dwóch cyfr po prawej stronie
236	wiersz 21 od góry	3. Modyfikacji wymaga	4. Modyfikacji wymaga
236	wiersz 8 od góry	Ten przykład jest trochę	3. Ten przykład jest trochę
238	wiersz 8 od dołu	/^(\S+)\s.*\1$/	/^(\S+)\s.*\s$/
239	wiersz 19 od góry	za nawisem	a nawiasem
249	wiersz 3 od góry	G	g
271	wiersz 1 w ramce	dodarcie	dotarcie
275	wiersz 5 od dołu	statystka	statystyka

276	wiersz 9 od góry	w listingu 11.1.	w listingu 11.2.
319	wiersz 10 od dołu	use Mojmodul wq($licznik)	use Mojmodul qw($licznik)
319	wiersz 2 od dołu	use Mojmodul sq(elem1, elem3)	use Mojmodul qw(elem1, elem3)
633	wiersz 10 od góry	Potrzenby	Potrzebny

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

Strona	Linia	Jest	Powinno
103	**27**	**Do pętli foreach wrócimy w rozdziale szóstym podczas omawiania tablic rozproszonych.**	**Do pętli foreach wrócimy w rozdziale piątym podczas omawiania tablic rozproszonych.**
173	**15**	**if ($wyjscie ne ´y´ && $wyjscie ne ´n´) {**	**if ($wyjscie ne ´t´ && $wyjscie ne ´n´) {**