

Narration	Time
Including files or modules پر اسپوکن ٹیوٹوریل میں خوش آمدید۔	00:01
اس ٹیوٹوریل میں ہم Perl پروگرامنگ میں مندرجہ ذیل میٹھڈس استعمال کرنا سیکھیں گے: do use اور require	00:08
اس ٹیوٹوریل کے لئے میں: انٹ لینکس 12.04 آپریٹنگ سسٹم Perl 5.14.2 اور گڈیٹ ٹیکسٹ ایڈیٹر استعمال کر رہا ہوں	00:16
آپ اپنی پسند کا کوئی بھی ٹیکسٹ ایڈیٹر استعمال کر سکتے ہیں۔	00:28
اس ٹیوٹوریل کو سمجھنے کے لئے آپ کو Perl پروگرامنگ معلومات ہونی چاہئے۔	00:32
اگر نہیں تو متعلقہ Perl اسپوکن ٹیوٹوریل کے لئے اسپوکن ٹیوٹوریل ویب سائٹ پر جائیں۔	00:37
do method (): یہ موجودہ سکرپٹ فائل میں دوسری فائلوں سے سورس کوڈ کو داخل کرنے کے آسان طریقے ہیں۔	00:44
اب سمجھتے ہیں کہ do()method کو کیسے استعمال کرتے ہیں۔	00:53
آپ ٹیکسٹ ایڈیٹر میں نئی فائل کھولیں اور اسے datetime dot pl نام دیں۔	00:57
datetime dot pl فائل میں، سکرین پر دکھائے گئے طریقے سے مندرجہ ذیل کوڈ ٹائپ کریں۔	01:03
یہاں سے آگے، ٹرمنل پر ہر کمانڈ کے بعد Enter کی دبا نایا درکھیں۔	01:09
اب کوڈ سمجھتے ہیں۔	01:15
موجودہ ڈیٹ اور ڈائنامک dollar datestring ویریبل میں جمع کیا جاتا ہے۔	01:18
یہاں، میرے پاس msgThanks نامی ایک فنکشن ہے جو ایک Thank you میسج رٹرن کرتا ہے۔	01:23
فائل سیو کرنے کے لئے Ctrl + S دبائیں۔	01:31
آگے، ایک اور Perl پروگرام دیکھتے ہیں جو اس فائل datetime dot pl کو استعمال کرے گی۔	01:35
آپ ٹیکسٹ ایڈیٹر میں ایک نئی فائل کھولیں اور اسے main dot pl نام دیں۔	01:43

01:49	main dot pl فائل میں، سکریں پر دکھائے گئے طریقے سے مندرجہ ذیل کو ڈٹائپ کریں۔
01:55	اب میں کوڈ سمجھاتا ہوں۔
01:58	یہاں، پہلی لائن ویلیم مسیج پرنٹ کرتی ہے۔
02:03	'do()' میٹھڈ اس فائل کے نام کے ساتھ کال ہوتا ہے جہاں سے ہم کوڈ استعمال کرنا چاہتے ہیں۔
02:09	موجودہ ڈیٹ اور ٹائم file datetime dot pl کے datestring \$ ویریبل میں جمع کیا جاتا ہے۔
02:16	اور آخر میں، ہم () msgThanks فنکشن کو اسی فائل سے کال کرتے ہیں۔
02:21	اب، فائل کو سیو کرنے کے لئے Ctrl + S دبائیں۔
02:25	اب پروگرام کو ایکزیکوٹ کرتے ہیں۔
02:27	ٹرمنل پرواپس جائیں اور ڈٹائپ کریں: perl main dot pl اور اینٹر دبائیں۔
02:34	ٹرمنل پر آؤٹ پٹ کے لیے دیکھیں۔
02:37	آگے ہم سیکھیں گے کہ Perl پروگرام میں require() میٹھڈ اور use() میٹھڈ کو کس طرح استعمال کرتے ہیں۔
02:44	یہ میٹھڈس تب استعمال ہوتے ہیں جب ہمارے پاس subroutines کا مجموعہ ہوتا ہے جو ایک سے زیادہ Perl پروگراموں میں استعمال کئے جاسکتے ہیں۔
02:52	use() method صرف modules کے لئے استعمال کیا جاتا ہے۔
02:56	اسکی کمپائلیشن کے وقت تصدیق ہوتی ہے۔
02:59	فائل extension دینے کی کوئی ضرورت نہیں ہے۔
03:03	require() میٹھڈ، Perl پروگرام اور modules دونوں کے لئے استعمال کیا جاتا ہے۔
03:08	اس کی run ہوتے وقت تصدیق ہوتی ہے۔
03:10	فائل ایکسٹینشن دینے کی ضرورت ہوتی ہے۔
03:14	use میٹھڈ کا سنٹیکس ہے use module name semicolon
03:20	Perl modules وہ فائلیں ہیں جو .pm ایکسٹینشن سے ختم ہوتی ہیں۔
03:25	کوڈ کے دوبارہ استعمال کا عمل، modules کے ذریعے لاگو کیا جاتا ہے۔
03:30	یہ دیگر (کمپیوٹر) لینگویجز کی libraries جیسی ہی ہوتی ہیں۔
03:35	اب، میں Perl کوڈ میں ماڈیول کو داخل کرنے کے لئے use method کے ساتھ سادہ پروگرام دکھاؤں گا۔

03:43	آپ ٹیکسٹ ایڈیٹر میں ایک نئی فائل کھولیں اور اسے sum dot pm نام دیں۔
03:49	sum dot pm فائل میں، سکرین پر دکھائے گئے طریقے سے کوڈ ٹائپ کریں۔
03:55	یہاں، میرے پاس ایک سادہ function ہے جو دیے گئے نمبرس کے سیٹ کے جوڑ کا حساب کرے گا۔
04:01	فائل سیو کرنے کے لئے Ctrl + S دبائیں۔
04:05	ہم ایک دوسرا Perl سکرپٹ لکھیں گے جہاں ہم یہ sum dot pm فائل استعمال کریں گے۔
04:11	اب میں سیمپل پروگرام app dot pl کھولتا ہوں جو میں نے پہلے ہی سیو کر لیا ہے۔
04:17	app dot pl فائل میں، سکرین پر دکھائے گئے طریقے سے مندرجہ ذیل کوڈ ٹائپ کریں۔
04:22	اب میں کوڈ سمجھاتا ہوں۔
04:25	پہلی لائن ماڈیول نام کے ساتھ use method ظاہر کرتا ہے۔
04:29	ہمارے کیس میں، ماڈیول کا نام sum ہے۔
04:33	ہم sum dot pm فائل میں total () فنکشن پر ان پٹ کے پیرامیٹرز کی طرح 1، 4، 5، 7، 9 پاس کر رہے ہیں۔
04:44	دوبارہ، اگلی لائن میں، ہم اسی فنکشن پر 1 سے 10 تک ان پٹ کے پیرامیٹرز پاس کر رہے ہیں۔
04:52	اب فائل کو سیو کرنے کے لئے Ctrl + S دبائیں۔
04:56	اب پروگرام کو ایکزیکوٹ کریں۔
04:59	ٹرینل پرواپس جائیں اور ٹائپ کریں: perl app dot pl: اور اینٹر دبائیں۔
05:06	ٹرینل پر دکھائے گئے آؤٹ پٹ کو دیکھیں۔
05:10	اب use method میں کچھ مزید آپشنس دیکھتے ہیں۔ ٹیکسٹ ایڈیٹر میں sum dot pm پرواپس جائیں۔
05:18	سورس کوڈ کے آغاز میں مندرجہ ذیل لائنز شامل کریں use warnings ، semicolonuse strict semicolon
05:27	use strict اور use warnings ، compiler flags ہوتے ہیں جو Perl کو زیادہ سخت طریقے سے برتاؤ کرنے کے لئے ہدایات دیتے ہیں۔
05:35	یہ عام پروگرامنگ غلطیوں کو روکنے کے لئے استعمال ہوتے ہیں۔
05:39	use strict ، پروگرام میں استعمال ہوئے سارے وریبلز کو ڈیکلئیر کرنے کے لئے یوزر کو مجبور کرتا ہے۔

05:45	اگر errors ہوتے ہیں تو use strict ایکڑ کیوشن کو ختم کر دے گا۔
05:50	use warnings صرف انتباہ دے گا لیکن ایکڑ کیوشن جاری رکھے گا۔
05:56	مانیں کہ ہم وریبل \$ sum کو my کی طرح ڈکلیئر کرنا بھول گئے۔
06:02	اب دیکھتے ہیں کہ یہی پروگرام کس طرح ایکڑ کیوٹ ہوتا ہے۔
06:06	فائل سیو کرنے کے لئے Ctrl + S دبائیں۔
06:09	ٹرمینل پر جائیں اور ٹائپ کریں: perl app dot pl
06:15	ہم دیکھ سکتے ہیں کہ نتیجے کی ایکڑ کیوشن کے بغیر ہی پروگرام ختم ہو جاتا ہے۔
06:21	ٹرمینل پر دکھائی گئی لائنوں کا پہلا سیٹ use strict کی طرف سے بنائی ہوئی error messages ہیں۔
06:29	آخری دو abort میسیجس ہیں۔
06:32	تو اس طرح use method آپشن کام کرتا ہے۔
06:36	آگے اب ہم ایک Perl پروگرام دیکھیں گے، جہاں ہم require مہتھڈ استعمال کرتے ہیں۔
06:41	اب میں ایک سیمپل پروگرام common functions dot pl کھولتا ہوں جو میں نے پہلے ہی سیو کر لیا ہے۔
06:48	آپ common functions dot pl فائل میں سکرین پر ظاہر طریقے سے مندرجہ ذیل کوڈ ٹائپ کریں۔ اب کوڈ سمجھتے ہیں۔
06:57	یہاں، ہم عام طور پر استعمال ہوئے functions کا مجموعہ دیکھ سکتے ہیں۔
07:01	پہلا فنکشن square ()، نمبر کا اسکوائر رٹرن کرتا ہے۔
07:06	دوسرا فنکشن square underscore root ()، دئے گئے نمبر کا اسکوائر رُوٹ رٹرن کرتا ہے۔
07:12	اگلا فنکشن random underscore number () ایک رینڈم نمبر پیدا کرتا ہے۔
07:18	آخری فنکشن random underscore range ()، نمبرس کی lower یعنی نچلی ریج اور upper یعنی اوپری ریج کے درمیان ایک رینڈم نمبر پیدا کرتا ہے۔
07:26	نوٹ کریں کہ ہمیں فائل کے آخر میں 1 semicolon (;) کی ضرورت ہے۔
07:31	یہ اس لئے ہے کیونکہ Perl کو ایک true ویلیور یٹرن کرنے کے لئے فائل میں آخری ایکسپریشن کی ضرورت ہے۔
07:37	فائل سیو کرنے کے لئے Ctrl + S دبائیں۔

07:41	اب ہم ایک Perl پروگرام لکھیں گے جس میں ہم require میتھڈ استعمال کرتے ہوئے ان subroutines کو کال کریں گے۔
07:48	اب میں وہ سیمپل پروگرام call program dot pl کھولتا ہوں جو میں نے پہلے ہی سیکھا ہے۔
07:54	اپنی فائل میں سکرین پر ظاہر مندرجہ ذیل کوڈ ٹائپ کریں . اب میں کوڈ سمجھاتا ہوں۔
08:02	Perl require ، کوڈ رکھنے والی common functions dot pl فائل پڑھتا ہے اور اسے کمپائل کرتا ہے۔
08:09	یہ پروگرام یوزر کو 4 آپشنس دیتا ہے . یوزر کو ایک بار میں ایک ہی آپشن منتخب کرنا ہے۔
08:17	1: ایک ، نمبر کا اسکور حاصل کرنے کیلئے۔
08:20	2: دو ، نمبر کے سکورز روٹ کے لئے ہے۔
08:23	3: تین ، دی گئی رینج میں رینڈم (بے ترتیب) تعداد کے لئے ہے
08:29	4: چار ، پروگرام سے باہر آنے کے لئے ہے۔
08:29	اگر آپشن 1 ٹائپ کیا جاتا ہے تو یہ یوزر سے ایک نمبر درج کرنے کو کہے گا۔
08:34	ویلیو \$number میں جمع ہوتی ہے . ویلیو commonfunctions dot pl فائل میں square () فنکشن کو پاس کی جاتی ہے۔
08:44	function نمبر کے سکورز کو رٹرن کرتا ہے۔
08:47	پرنٹ اسٹیٹمنٹ آؤٹ پٹ کی طرح نمبر کے سکورز کو پرنٹ کرتا ہے۔
08:52	اگر آپشن 2 ٹائپ کیا جاتا ہے تو آؤٹ پٹ میں نمبر کا سکورز روٹ ظاہر ہوتا ہے۔
08:58	گزشتہ فنکشن square () میں سمجھائے گئے طریقے سے ایکزیکوشن کی جاتی ہے۔
09:03	اگر آپشن 3 ٹائپ کیا جاتا ہے تو دی گئی رینج میں آؤٹ پٹ کے طور پر ایک رینڈم (بے ترتیب) نمبر ظاہر ہوتا ہے۔
09:09	دوسری صورت میں اگر آپشن 4 ہوتا ہے تو پروگرام exit ہو جاتا ہے . اگر بیان کردہ آپشنس کے علاوہ کوئی اور آپشن دیا جاتا ہے تو پرنٹ اسٹیٹمنٹ دکھاتا ہے Incorrect option .
09:20	نوٹ کریں اس پروگرام میں ہم نے commonfunctions dot pl سے چار میں سے صرف تین فنکشنس کال کئے ہیں۔
09:28	فائل سیو کرنے کے لئے Ctrl + S دبائیں۔
09:31	پروگرام کو ایکزیکوٹ کریں۔

09:34	ٹریبل پرواپس جائیں اور ٹائپ کریں: perl callprogram dot pl:
09:41	آؤٹ پٹ ملاحظہ کریں۔
09:44	میں ایک بار پھر مختلف آپشنس کے ساتھ پروگرام کو ایکزیکوٹ کروں گا۔
09:49	ٹائپ کریں: perl callprogram dot pl.
09:53	اب آپشن کے طور پر 3 درج کریں
09:56	Enter a lower range میں درج کریں 50
09:59	Enter a upper range میں درج کریں 99
10:02	ہم دیکھ سکتے ہیں کہ نمبرس کی دی گئی رینج میں رینڈم (بے ترتیب) نمبر پیدا ہوتا ہے۔
10:08	اپنے آپ دوسرے آپشنس کی کوشش کریں۔
10:11	یہ ہمیں اس ٹیوٹوریل کے آخر میں لاتا ہے۔ اس کا خلاصہ بیان کرتے ہیں
10:16	اس ٹیوٹوریل میں ہم نے Perl پروگرامنگ میں مندرجہ ذیل میٹھڈس کو استعمال کرنا سیکھا: do use require
10:24	نوٹ: require: ماڈیول کی جگہ use ماڈیول کی صلاح دی جاتی ہے کیونکہ یہ compile time پر ماڈیول کی دستیابی کا تعین کرتا ہے۔
10:33	یہاں آپ کے لئے ایک مشتق ہے۔ ایک Perl پروگرام reminder.pl لکھیں، جہاں آپ شرکاء کو ایک خط لکھیں گے۔
10:41	یوزر کو To اور From نام داخل کرنے کے لئے کہیں۔
10:45	use میٹھڈ استعمال کرتے ہوئے Letter dot pm سے subroutines کو کال کریں۔
10:50	Letter dot pm فائل میں نیچے والے فنکشنس لکھیں۔
10:54	LetterDate () فنکشن موجودہ ڈیٹ اور ٹائم کو رٹرن کرتا ہے۔
10:58	To () فنکشن شرکاء کے نام رٹرن کرتا ہے۔
11:02	From () فنکشن بھیجنے والے کا نام رٹرن کرتا ہے۔

11:05	Lettermsg () فنکشن خط کے موضوع اشیاء کو واپس کرتا ہے۔
11:09	Thanksmsg () فنکشن thanks اور regards واپسی کرتا ہے۔
11:13	آؤٹ پٹ یہاں دکھائی دے رہے جیسا نظر آنا چاہئے۔
11:20	مندرجہ ذیل لنک پر دستیاب ویڈیو اسپوکن ٹیوٹوریل پروجیکٹ کا خلاصہ کرتا ہے۔ اسے ڈاؤن لوڈ کریں اور دیکھیں۔
11:27	اسپوکن ٹیوٹوریل پروجیکٹ ٹیم اسپوکن ٹیوٹوریلز کا استعمال کرتے ہوئے ورکشاپ چلاتی ہے اور آن لائن ٹیسٹ پاس کرنے والوں کو سند دیتے ہیں۔
11:36	مزید معلومات کے لئے ہمیں لکھیں۔
11:40	اسپوکن ٹیوٹوریل پروجیکٹ حکومت ہند کے ایچ آر ڈی کے NMEICT طرف حمایت شدہ ہے۔ اس مشن پر مزید معلومات اس لنک پر دستیاب ہیں۔
11:51	آئی آئی ٹی بائے سے میں وجاحت احمد آپ سے رخصت لیتا ہوں۔ شامل ہونے کیلئے آپ کا شکریہ۔