Narration	Time
'Perl' میں 'Referencing and Dereferencing' پر اسپوکن ٹیوٹوریلس میں خوش آمدید.	00:01
اس ٹیوٹوریل میں ہم مندرجہ ذیل کے بارے میں سیکھیں گے: 'Scalar References' 'Array References' 'Hash References' 'Dereferences' اور array/hash references کے elements کو کیسے شامل کرتے، ہٹاتے، اور رسائی کرتے ہیں .	00:07
اس ٹیوٹوریل کے لئے، میں: 'Ubuntu Linux 12.04 آپریٹنگ سسٹم 'Perl 5.14.2' اور 'gedit ٹیکسٹ ایڈیٹر' استعمال کر رہا ہوں	00:22
آپ اپنی پسند کا کوئی بھی ٹیکسٹ ایڈیٹر استعمال کر سکتے ہیں.	00:33
آپ کو مندرجہ ذیل کی ورکنگ یعنی فعالی معلومات ہونی چاہیے: 'Perl' پروگرامنگ Array functions اور Hash functions	00:37
اگر نہیں تو اس ویب سائٹ پر دستیاب متعلقہ 'Perl' ٹیوٹوریلس کو دیکھیں.	00:43
'Reference' کیا ہوتی ہے؟	00:49
ایک 'reference'، ایک 'variable، array، hash' یا ایک 'subroutine' کا pointer یا address ہوتی ہے.	00:51
یہ براہ راست ڈیٹا نہیں رکھتی ہے.	00:58
'Reference' ایک آسان، گھنے scalar ویلیو ہوتی ہے.	01:01
'Reference'، Perl کوڈ کی کارکردگی کو سدھارے گی جب آپ بڑے ڈیٹا سٹرکچرس کو پاس یا رِٹرن کرتے ہیں.	01:05

یہ میموری بچاتی ہے کیونکہ ایک ویلیو پاس کرنے کے بجائے یہ subroutine پر reference پاس کرتی ہے.	01:12
پرل کے پیچیدہ ڈیٹا سٹرکچرس کو آسانی سے مینیج کرتی ہے.	01:18
اب سیکھتے ہیں کہ ایک 'reference' کو کیسے بناتے ہیں.	01:22
ہم اس کے آگے ایک (\) backslash لگا کر کسی بھی variable, subroutine یا value کے لئے reference بنا سکتے ہیں.	01:25
ایک scalar variable، یہاں ظاہر طریقے backslash اور ($) dollar sign سے reference کیا جاتا ہے.	01:33
ایک array variable، backslash اور '(@) at the rate' سمبل یعنی علامت سے reference کیا جاتا ہے.	01:39
ایک hash variable یہاں مثال میں ظاہر طریقے پر backslash اور (%)percentage سمبل سے reference کیا جاتا ہے.	01:45
'dereference' کیا ہوتا ہے؟	01:53
جب ایک reference، dereference کیا جاتا ہے، تو حقیقی ویلیو ریٹرن ہوتی ہے.	01:55
'Dereference'، reference ویریبل کو کرلی بریکٹ میں بند کر کے کیا جاتا ہے	02:00
اور کرلی بریکٹ سے پہلے اس reference کی قسم بتانے والا کریکٹر آتا ہے.	02:06
اب دیکھتے ہیں کہ ویریبلس کو dereference کس طرح کرتے ہیں.	02:12
ایک scalar ویریبل ڈالر سائن اور کرلی بریکیٹس سے dereference کیا جاتا ہے.	02:16
ایک array ویریبل '(@) at the rate' سمبل اور کرلی بریکیٹس سے dereference کیا جاتا ہے.	02:21
ایک hash variable، (%)percentage سمبل اور کرلی بریکیٹس سے dereference کیا جاتا ہے.	02:27
اب Scalar reference اور dereference کے لئے ایک سادہ پروگرام دیکھتے ہیں.	02:33
اب میں 'gedit' ٹیکسٹ ایڈیٹر میں ایک سیمپل پروگرام کھولتا ہوں.	02:38
terminal کھولیں اور ٹائپ کریں: gedit scalarRef dot pl ampersand اور اینٹر دبائیں.	02:43
سکرین پر دکھائے گئے طریقے سے مندرجہ ذیل کوڈ ٹائپ کریں.	02:50

02:55	اب میں کوڈ سمجھاتا ہوں.
02:57	پہلی لائن ایک scalar variable ، '$ a' ڈکلئیر کرتی ہے اور 10 سے انشلائز کرتی ہے.
03:03	جیسا کہ پہلے بتایا گیا ہے کہ سکیلر ویریبل backslash اور ڈالر سائن سے ریفرنس کیا جاتا ہے.
03:10	یہ لائن، ویریبل کا memory address پرنٹ کرے گی جو reference کی طرح بنایا جاتا ہے.
03:16	حقیقی ویلیو کو پرنٹ کرنے کے لئے، variable کرلی بریکٹ سے پہلے '$' سمبل لگا کر dereference کیا جاتا ہے.
03:23	یہاں '()ref' فنکشن، 'reference type' جیسے scalar یا array یا hash رِٹرن کرے گا.
03:30	اب، سیو کرنے کے لئے 'Ctrl + S' دبائیں.
03:34	اب میں پروگرام کو ایکزِ کیوٹ کرتا ہوں.
03:36	terminal پر جائیں اور ٹائپ کریں: perl scalarRef dot pl اور اینٹر دبائیں.
03:43	آؤٹ پٹ ایسا لگتا ہے.
03:46	پہلی لائن memory address ظاہر کرتی اہے جہاں ویلیو 10 جمع کیا جاتا ہے.
03:51	دوسری لائن حقیقی ویلیو 10 ریٹرن کرتی ہے.
03:55	'()Ref' فنکشن، آؤٹ پُٹ میں 'SCALAR' رِٹرن کرتا ہے.
03:59	آگے، سمجھتے ہیں کہ ایک سیمپل پروگرام استعمال کرتے ہوئے reference اور dereference، array کو کیسے بناتے ہیں.
04:07	میرے پاس پہلے سے سیمپل پروگرام ہے . میں اسے 'gedit' ٹیکسٹ ایڈیٹر میں کھولتا ہوں.
04:13	terminal پر ٹائپ کریں : gedit arrayRef dot pl ampersand اور اینٹر دبائیں.
04:20	arrayRef dot pl فائل میں سکرین پر ظاہر طریقے پر کوڈ ٹائپ کریں.
04:26	اب میں کوڈ سمجھاتا ہوں.
04:28	یہاں پہلی لائن میں، میں نے color@ نامی ایک array ڈکلئیر کیا ہے اور اسے 3 ویلیو سے انشلائز کیا ہے.
04:35	یہ color@ backslash سے ریفرنس کیا جاتا ہے جو ارے-نیم ہے اور ' colorRef $ ' کو تفویض کیا جاتا ہے.
04:42	'print' کا اسٹیٹمینٹ reference ویلیو اور dereferenced ویلیو کو پرنٹ کرے گا.

04:47	اب فائل کو سیو کرنے کے لئے 'Ctrl + S' دبائیں.
04:51	اب ہم پروگرام کو ایکزِکیوٹ کرتے ہیں.
04:53	'ٹرمنل' پر واپس جائیں اور ٹائپ کریں : perl arrayRef dot pl اور اینٹر دبائیں.
05:00	آؤٹ پٹ اس طرح کا لگتا ہے.
05:04	پہلی لائن ویریبل کے memory address کا آؤٹ پٹ دکھاتی ہے جو reference کی طرح بنائی جاتی ہے.
05:10	دوسری لائن حقیقی ویلیو دکھاتی ہے جو dereference کی جاتی ہے.
05:16	آگے ہم دیکھیں گے کہ ایک array کے لئے direct reference کو کیسے ڈکلئیر کرتے ہیں.
05:21	اب اپنے پروگرام پر واپس آتے ہیں.
05:24	میں نے ایک array کے 'direct reference' کو دکھانے کے لئے موجودہ پروگرام کو تبدیل کر دیا ہے.
05:29	آپ دکھائے گئے طریقے سے چوکور بریکیٹس [] استعمال کر کے ایک array کے لئے 'direct reference' بنا سکتے ہیں.
05:35	dereference کرنے کے لئے (->) arrow operator استعمال کریں.
05:39	'print' اسٹیٹمینٹ آؤٹ پٹ میں 'Green' پرنٹ کرے گا.
05:43	یہاں، 'print' اسٹیٹمینٹ انڈیکس آف (1) کی ویلیو لیتا ہے جو کہ ہمارے پروگرام میں 'Green' ہے.
05:50	فائل سیو کرنے کے لئے 'Ctrl + S' دبائیں.
05:54	'ٹرمنل' پر واپس جائیں اور ٹائپ کریں : perl arrayRef dot pl اور ایکزِکیوٹ کرنے کے لئے اینٹر دبائیں.
06:03	میں ایک مثال دکھاوں گا کہ اِسی کوڈ فائل میں 'direct hash reference' کس طرح استعمال کرتے ہیں. لہذا، 'gedit' پر جائیں.
06:11	آپ یہاں ظاہر طریقے کی طرح کرلی بریکٹ '()' استعمال کرتے ہوئے ہیش پر direct reference بنا سکتے ہیں.
06:18	اسے dereference کرنے کے لئے (->) arrow operator استعمال کریں۔ 'Name'، ہیش کی ہے.
06:24	کوڈ کے اس بلاک کو ایکزِکیوٹ کرنے پر دونوں print اسٹیٹمینٹس، آؤٹ پُٹ میں 'Sunil' پرنٹ کریں گے.

06:31	آ گے ہم دیکھیں گے کہ سیمپل پروگرام کے ساتھ array reference پر elements کو کیسے شامل کرتے، مٹاتے اور رسائی کرتے ہیں.
06:39	میرے پاس پہلے سے ایک سیمپل پروگرام ہے .میں اسے 'gedit' ٹیکسٹ ایڈیٹر میں کھولتا ہوں.
06:45	terminal کھولیں اور ٹائپ کریں : gedit arrayRefadd dot pl ampersand اور اینٹر دبائیں.
06:54	'arrayRefadd.pl' فائل اب 'gedit' میں کھلتی ہے .اپنی فائل میں یہاں ظاہر طریقے سے کوڈ ٹائپ کریں.
07:02	پہلی لائن array کو انشلائز کرتی ہے.
07:06	ہم نے 'backslashnumarray' سے array کو reference کیا ہے اور '$ ref' کو تفویض کیا ہے.
07:13	اب، ہم دیکھیں گے کہ array reference سے ایک خصوصی element تک کیسے رسائی کرتے ہیں.
07:19	ہمیں کسی مخصوص ویلیو تک رسائی کرنے کے لئے اسکوائر بریکٹ[] میں array index کو اور اسے dereference کرنے کے لئے(”->“) arrow operator استعمال کرنے کی ضرورت ہے.
07:28	print اسٹیٹمینٹ ،انڈیکس آف زیرو'(0) ' کی ویلیو پرنٹ کرے گا.
07:32	'Push ()' فنکشن array reference کی آخری پوزیشن پر elements کو جوڑتا ہے. ہمارے کیس میں موجودہ array 1، 2، 3، 4 کے آخر میں 5، 6، 7 جوڑے جاتے ہیں .
07:47	یہ print اسٹیٹمینٹ array reference پر شامل کرنے کے بعد آؤٹ پٹ ظاہر کرتا ہے.
07:53	'Pop ()' فنکشن، array reference کی آخری پوزیشن سے element کو خارج کرتا ہے.
07:58	ہماری مثال میں، 7 موجودہ array reference سے ہٹا دیا جائے گا.
08:03	array reference سے مٹانے کے بعد print اسٹیٹمینٹ آؤٹ پُٹ ظاہر کرتا ہے.
08:08	اب، فائل سیو کرنے کے لئے' لئے 'Ctrl + S دبائیں.
08:11	اب پروگرام کو ایکزِ کیوٹ کرتے ہیں.
08:14	'ٹرمنل 'پر جائیں اور ٹائپ کریں : perl arrayRefadd dot pl اور اینٹر دبائیں.
08:22	آؤٹ پٹ اس طرح کا لگتا ہے.
08:26	اب، hash reference کے elements کو شامل کرنے، ہٹانے اور اُن تک رسائی کے لئے ایک اور سیمپل پروگرام دیکھتے ہیں.

08:34	'ٹرمنل 'پر، ٹائپ کریں : gedit hashRefadd dot pl ampersand اور اینٹر دبائیں.
08:42	یہ 'gedit'میں 'hashRefadd.pl' فائل ظاہر کرے گا.
08:47	اب میں سیمپل پروگرام سمجھاتا ہوں.
08:50	میں نے 'direct hash reference'ڈکلئیر کیا ہے جو scalar ویریبل '$ weektemp'میں جمع کیا جا سکتا ہے.
08:57	میں نے hash reference کو دکھانے کے لئے' کرلی بریکٹ 'اور dereference کے لئے arrow operatorاستعمال کیا ہے.
09:04	یہ کوڈ منڈے سے جمعہ تک کے ٹیمپریچر یعنی درجہ حرارت کی ویلیوز کو جمع کرتا ہے.
09:09	میں ہیش کی کیز میں لوپ کرنے کے لئے "keys" built-in function استعمال کر رہا ہوں.
09:15	printاسٹیٹمینٹ ہیش کے ہر ایلیمیٹ کو پرنٹ کرے گا.
09:19	ہم یہاں ظاہر طریقے سے element کی خصوصی ویلیو تک رسائی حاصل کر سکتے ہیں.
09:25	printاسٹیٹمینٹ منڈے کا ٹیمپریچر پرنٹ کرے گا.
09:29	اب فائل کو سیو کریں.
09:32	'ٹرمنل 'پر جائیں اور ٹائپ کریں:perl hashRefadd dot pl اور آؤٹ پٹ دیکھنے کے لئے اینٹر دبائیں.
09:41	hash keysاور hash values رینڈم یعنی بلاترتیب آرڈر میں جمع کی جاتی ہیں.
09:46	ظاہر آؤٹ پٹ اس ترتیب کے مطابق نہیں ہوتا جس میں وہ شامل کئے گئے تھے.
09:52	اس کے ساتھ ہم اس ٹیوٹوریل کے آخر میں آ گئے ہیں . اب خلاصہ بیان کرتے ہیں.
09:57	اس ٹیوٹوریل میں ہم نے مثالوں سے مندرجہ ذیل کے بارے میں سیکھا: 'Scalar References' 'Array References' 'Hash References' 'Dereferences' اور مثالوں سمیت array/hash references کے elements کو کیسے شامل کرتے، ہٹاتے، اور رسائی کرتے ہیں .

یہاں آپ کے لئے ایک مشق ہے .ہماری 'hashRefadd dot pl file' میں، 'weektemp' ہیش میں نئی کیز 'Saturday' اور 'Sunday' شامل کریں.	10:14
آخر میں 'Saturday' حذف کریں.	10:24
ہیش 'weektemp' پرنٹ کریں.	10:27
پروگرام کو سیو اور ایکزِکیوٹ کریں .اب نتیجہ چیک کریں.	10:30
مندرجہ ذیل لنک پر دستیاب ویڈیو اسپوکن ٹیوٹوریل پروجیکٹ کا خلاصہ کرتا ہے .اسے ڈاؤن لوڈ کریں اور دیکھیں.	10:35
ہم ورکشاپ چلاتے ہیں اور آن لائن ٹیسٹ پاس کرنے والوں کو سند دیتے ہیں .مزید معلومات کے لئے ہمیں لکھیں.	10:42
اسپوکن ٹیوٹوریل پراجیکٹ، بھارتی حکومت کے ایم ایچ آر ڈی کے NMEICT کی طرف حمایت شدہ ہے. اس مشن پر مزید معلومات اس لنک پر دستیاب ہیں.	10:51
آئی آئی ٹی بامبے سے میں وجاحت احمد اب آپ سے رخصت لیتا ہوں .شامل ہونے کے لئے شکریہ.	11:02