

Array Functions

Talk to a Teacher

<http://spoken-tutorial.org>

National Mission on Education through ICT

<http://sakshat.ac.in>

Amol Brahmanekar

28 July 2013



Learning Objectives

In this tutorial, we will learn about
Array Functions in Perl



Learning Objectives

In this tutorial, we will learn about
Array Functions in Perl

- ▶ push
- ▶ pop
- ▶ shift



Learning Objectives

In this tutorial, we will learn about
Array Functions in Perl

- ▶ push
- ▶ pop
- ▶ shift
- ▶ unshift
- ▶ split
- ▶ splice



Learning Objectives

In this tutorial, we will learn about
Array Functions in Perl

- ▶ push
- ▶ unshift
- ▶ join
- ▶ pop
- ▶ split
- ▶ sort
- ▶ shift
- ▶ splice
- ▶ qw



System Requirements

- ▶ **Ubuntu Linux 12.04 OS**



System Requirements

- ▶ **Ubuntu Linux 12.04 OS**
- ▶ **Perl 5.14.2**



System Requirements

- ▶ **Ubuntu Linux 12.04 OS**
- ▶ **Perl 5.14.2**
- ▶ **gedit Text Editor**



Pre-requisites

- ▶ **Knowledge of Variables, Data-structures and Arrays**



Pre-requisites

- ▶ **Knowledge of Variables, Data-structures and Arrays**
- ▶ **Knowledge of comments, loops and conditional statements (added advantage)**



Pre-requisites

- ▶ Knowledge of Variables, Data-structures and Arrays
- ▶ Knowledge of comments, loops and conditional statements (added advantage)
- ▶ Please go through relevant spoken tutorials on <http://spoken-tutorial.org>



Inbuilt Functions

- ▶ Perl provides certain inbuilt functions



Inbuilt Functions

- ▶ Perl provides certain inbuilt **functions**
- ▶ These **functions** can perform operations on an **Array**



push-pop

To add and remove elements from the last position of an **Array**, use -



push-pop

To add and remove elements from the last position of an **Array**, use -

- ▶ **push** function: adds an element at the last position of an **Array**



push-pop

To add and remove elements from the last position of an **Array**, use -

- ▶ **push** function: adds an element at the last position of an **Array**
- ▶ **pop** function: removes an element from the last position of an **Array**



Syntax for push function

push function takes 2 arguments -



Syntax for push function

push function takes 2 arguments -

- ▶ **1st argument** is the **Array** in which to add an element



Syntax for push function

push function takes 2 arguments -

- ▶ **1st argument** is the **Array** in which to add an element
- ▶ **2nd argument** is the element which is to be **pushed** into the **Array**



Syntax for pop function

pop function takes only 1 argument -



Syntax for pop function

pop function takes only 1 argument -

- ▶ it is the **Array** from which an element needs to be removed



Note

Both these functions **push** and **pop** work at the last position of an **Array**



Note

The element removed by **pop** function can be collected into another variable



Note

The element removed by **pop** function can be collected into another variable

- ▶ The syntax for this is -



The element removed by **pop** function can be collected into another variable

- ▶ The syntax for this is -
- ▶ *`$variable = pop(@myArray)`*



shift-unshift

To add and remove an element from first position of an **Array** use -



shift-unshift

To add and remove an element from first position of an **Array** use -

- ▶ **unshift** function: adds an element to an **Array** at the 1st position



shift-unshift

To add and remove an element from first position of an **Array** use -

- ▶ **unshift** function: adds an element to an **Array** at the 1st position
- ▶ **shift** function: removes first element from an **Array**



Syntax for unshift

unshift function takes 2 arguments -



Syntax for unshift

unshift function takes 2 arguments -

- ▶ **1st argument** is the **Array** in which to add an element



Syntax for unshift

unshift function takes 2 arguments -

- ▶ **1st argument** is the **Array** in which to add an element
- ▶ **2nd argument** is the element to be added into the **Array**



Syntax for shift

shift function takes only one **argument**



Syntax for shift

shift function takes only one argument

- ▶ This is the **Array** from which the element needs to be removed



Note

Both these functions **unshift** and **shift** work at the first position of an **Array**



Note

We can collect the elements removed by **shift** function into some variable



Note

We can collect the elements removed by **shift** function into some variable

- ▶ The syntax for this is -



We can collect the elements removed by **shift** function into some variable

- ▶ The syntax for this is -
- ▶ *`$variable = shift(@myArray)`*



splice function

- ▶ **splice function** removes an element from a specified position of an **Array**



splice function

- ▶ **splice function** removes an element from a specified position of an **Array**
- ▶ Return value is an **Array** of removed elements



split function

- ▶ **split function** is used to divide a string at a specified delimiter



split function

- ▶ **split function** is used to divide a string at a specified delimiter
- ▶ Return value of this function is an **Array**



split function

- ▶ **split function** is used to divide a string at a specified delimiter
- ▶ Return value of this function is an **Array**
- ▶ Elements of this **Array** are the divided portions of the string



join function

join function

- ▶ joins the element of an **Array**, using specified delimiter



join function

join function

- ▶ joins the element of an **Array**, using specified delimiter
- ▶ Returns a string of joined elements



sort

sort function

- ▶ sorts the **Array** in alphabetical/numerical order



qw function

qw function

- ▶ returns an **Array** of words, separated by a white space



Syntax for split

split function takes 2 arguments -



Syntax for split

split function takes 2 arguments -

- ▶ 1st is the delimiter by which the string needs to be split



Syntax for split

split function takes 2 arguments -

- ▶ 1st is the delimiter by which the string needs to be split
- ▶ 2nd is the string which needs to be split



Syntax for split

split function takes 2 arguments -

- ▶ 1st is the delimiter by which the string needs to be split
- ▶ 2nd is the string which needs to be split
- ▶ delimiter can be specified in forward slash, single or double quotes



Syntax for join

join function takes 2 arguments -



Syntax for join

join function takes 2 arguments -

- ▶ 1st is the delimiter by which **Array** elements needs to be joined



Syntax for join

join function takes 2 arguments -

- ▶ 1st is the delimiter **by which Array** elements **needs to be joined**
- ▶ 2nd is the **Array**



Syntax for join

join function takes 2 arguments -

- ▶ 1st is the delimiter **by which Array** elements **needs to be joined**
- ▶ 2nd is the **Array**
- ▶ delimiters **can be specified in** single or double quotes



Syntax for sort

- ▶ **sort function** takes a single argument



Syntax for sort

- ▶ **sort function** takes a single argument
- ▶ which is the **Array** that needs to be sorted



Syntax for qw

qw function

- ▶ returns an **Array** of words, separated by space



Syntax for qw

qw function

- ▶ returns an **Array** of words, separated by space
- ▶ is not necessary to specify the word in "quotes", if written using **qw**



Summary

In this tutorial we learnt to -

- ▶ add/remove elements from an **Array**
- ▶ basic functions which can be performed on **Arrays**



Assignments

- ▶ 'script.spoken-tutorial.org/index.php/Perl'
- ▶ split the above string at '/'
delimiter
- ▶ Add https:// at the start of the
new **Array**
- ▶ Remove element 'Perl' from the
Array
- ▶ Declare a number **Array** and sort it



About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarises the Spoken Tutorial project



About the Spoken Tutorial Project

- ▶ Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarises the Spoken Tutorial project
- ▶ If you do not have good bandwidth, you can download and watch it



Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- ▶ Conducts workshops using spoken tutorials
- ▶ Gives certificates to those who pass an online test
- ▶ For more details, please write to contact@spoken-tutorial.org



Acknowledgements

- ▶ Spoken Tutorial Project is a part of the Talk to a Teacher project
- ▶ It is supported by the National Mission on Education through ICT, MHRD, Government of India
- ▶ More information on this Mission is available at

<http://spoken-tutorial.org/NMEICT-Intro>

