

# if-elsif-else, switch statements in Perl

**Talk to a Teacher**

**<http://spoken-tutorial.org>**

**National Mission on Education through ICT**

**<http://sakshat.ac.in>**

**Amol Brahmanekar**

**1 May 2013**



# Learning Objective

**In this tutorial, we will learn**



# Learning Objective

In this tutorial, we will learn

- ▶ **if-elsif-else** statement



# Learning Objective

In this tutorial, we will learn

- ▶ **if-elsif-else** statement
- ▶ **switch** statement in Perl



# System Requirements



# 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 & Comments**



# Pre-requisites

- ▶ **Knowledge of Variables & Comments**
- ▶ **Knowledge of for, foreach, while, do-while, if, if-else (added advantage)**



# Pre-requisites

- ▶ Knowledge of Variables & Comments
- ▶ Knowledge of for, foreach, while, do-while, if, if-else (added advantage)
- ▶ Please go through relevant spoken tutorials on

<http://spoken-tutorial.org>



# if-elsif-else Statement

- ▶ **if-elsif-else** statement can be used



# if-elsif-else Statement

- ▶ **if-elsif-else** statement can be used
- ▶ to test multiple conditions



# if-elsif-else Statement

- ▶ **if-elsif-else** statement can be used
- ▶ to test multiple conditions
- ▶ When all conditions fail



# if-elsif-else Statement

- ▶ **if-elsif-else** statement can be used
- ▶ to test multiple conditions
- ▶ When all conditions fail
- ▶ then it executes the default **else** block



# if-elsif-else statement syntax is

```
if (condition1) { piece of code; }  
    elsif (condition2)  
    { another piece of code; }  
    else  
    { code to be executed when both  
    the above conditions are false;  
    }
```





# if-elsif-else Statement

**The 3 cases imply that**



# if-elsif-else Statement

The 3 cases imply that

- ▶ Only one **if** block that satisfies the condition will be executed



# if-elsif-else Statement

The 3 cases imply that

- ▶ Only one **if** block that satisfies the condition will be executed
- ▶ otherwise the default **else** block will be executed



# if-elsif-else statement

We can have multiple **elsif** conditions as per our requirement



# if-elsif-else statement

We can have multiple **elsif** conditions as per our requirement

```
if (condition1) { code1; }  
    elsif (condition2) { code2; }  
    elsif (condition3) { code3; }  
    else { default-code; }
```



# Assignment

Write an **if-elsif-else** statement to print

- ▶ "I am a Science graduate" if stream is science
- ▶ "I am a Commerce graduate" if stream is commerce
- ▶ "I am an Arts graduate" if stream is not science or commerce



# switch Statement

- ▶ Till Perl 5.8, there was no **switch** in Perl



# switch Statement

- ▶ Till Perl 5.8, there was no **switch** in Perl
- ▶ After that, **Switch module** was introduced





# switch Statement

- ▶ Till Perl 5.8, there was no **switch** in Perl
- ▶ After that, **Switch module** was introduced
- ▶ which provided the functionality of **switch** statement



# switch statement syntax is

```
use Switch;  
switch ($value) {  
    case 1 {executes when $value=1}  
    case 'a' {exeutes when $value='a'}  
    else {executes when $value does not  
        match any of the cases}  
}
```



# switch statement

**The 3 cases imply that**



# switch statement

The 3 cases imply that

- ▶ The value of expression decides the **case** to be executed



# switch statement

The 3 cases imply that

- ▶ The value of expression decides the **case** to be executed
- ▶ Only the valid case will be executed



# switch statement

The 3 cases imply that

- ▶ The value of expression decides the **case** to be executed
- ▶ Only the valid case will be executed
- ▶ When there is no valid case, then the default **else** block will be executed



# switch statement

- ▶ It is not mandatory to write **else** case



# switch statement

- ▶ It is not mandatory to write **else** case
- ▶ In such a scenario,





# switch statement

- ▶ It is not mandatory to write **else** case
- ▶ In such a scenario,
  - ▶ if none of the cases match



# switch statement

- ▶ It is not mandatory to write **else** case
- ▶ In such a scenario,
  - ▶ if none of the cases match
  - ▶ then there will be no output from **switch** statement



# Assignment

Here is another assignment for you -

- ▶ Re-write the previous assignment
- ▶ given earlier in this tutorial
- ▶ using **switch** statement



# Summary

In this tutorial we learnt,

- ▶ **if-elsif-else** statement
- ▶ **switch** statement in Perl
- ▶ using sample programs



# About the Spoken Tutorial Project

- ▶ Watch the video available at [http://spoken-tutorial.org/What\\_is\\_a\\_Spoken\\_Tutorial](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](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](mailto: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>

