Perl

<https://www.perl.org/>

**Reading 1**

Read and make notes of the content in [Beginner's Introduction to Perl, Part 1](http://www.perl.com/pub/2000/10/begperl1.html) and read chapters 1 and 2 of [Learning Perl](http://docstore.mik.ua/orelly/perl4/lperl/index.htm) , also of [Beginner's Introduction to Perl, Part 2](http://www.perl.com/pub/2000/11/begperl2.html) and read chapters 4 and 6 of [Learning Perl](http://docstore.mik.ua/orelly/perl4/lperl/index.htm)

* Declare a variable that saves a number.
* Declare a variable that saves a sentence.
* Define an array that stores the split content of the sentence you just declared in different parts of the array.
* Add the number to the array.
* Print the results in console or shell
* Write a program that reads a file until the end (of the file, don’t be dramatic).
* The program should show all the words used in the file and the number of times each word was used.
* The words should be printed in alphabetic order.

**During class 1**

* What is Perl for?
* Why learn Perl?
* Python Vs Perl
* **Implement (basic strings and array examples.pl)**
* **Implement (loops and hashes.pl)**

**Lab1**

After doing this lab you will be able to perform simple scripts and automatize some tasks.

1. Write a program that reads a string and a number, and prints the string a number of times specified by the number in separate lines.
2. Write a program that asks for a number and a list of strings and then prints:
   1. The first string
   2. The last string
   3. The string indicated by the number
   4. A random string from the list. Use  *srand;* to set the random at the beginning of the program and *rand(@varArray)*  to choose an element from the list.
3. Write a program that reads a list of names of cities and prints them:
   1. Ordered
   2. Inverse order

The list should end with CTRL-D (unix) or CTRL-Z(win)

1. Write a program that ask for 3 numbers and writes:
   1. The lowest
   2. The highest
   3. “equal” if they’re all the same
2. Write a program that receives a list of numbers which includes the number 42 (for correctness). And then print the following:
   1. The sum of all the numbers in the list.
   2. The lowest one (don’t; use the sort function).
   3. The highest one (don’t; use the sort function).
   4. Every number below the 42.
3. Read this tutorial to set up a mail server or use an external sever, choose what best fits your needs. <http://www.tutorialspoint.com/perl/perl_sending_email.htm>

use this gmail account: [programming.languages.lab@gmail.com](mailto:programming.languages.lab@gmail.com) pwd: **mimamame**

or you can use your google account (but you need to allow the less secure mode in your google account):

<http://search.cpan.org/~lbrocard/Email-Send-Gmail-0.33/lib/Email/Send/Gmail.pm>

If you are on windows use the following command to install modules:

**ppm install <name>**

for example:

**ppm install MIME-Lite**

**ppm install Email::Send::Gmail**

Write a script that checks the date, opens a text file, and depending on the date, it will choose a personalized message for that holly day. The script must then send an email message to user. Add your own favourite dates at least 3 with your own personal messages. Choose your dates from <https://www.daysoftheyear.com/>

The text file can look something like this:

Valentine: Happy Valentine you are very, very, very special to me. Very, very much so. No, really I mean it. I wouldn’t write if it wasn’t so.

5 de Mayo: Happy independence my dear Mexican friend.

Merry Xmas: May you spend an awesome night with your awesome family in your awesome house. Awesome dude.

Emails:b323@somesever.com,a00882900@itesm.mx,bush@myownwhitehouse.org

**Reading 2**

Read and make notes of the content in [Beginner's Introduction to Perl, Part 3](http://www.perl.com/pub/2000/11/begperl3.html) and chapter 7 and 8 from [Learning Perl](http://docstore.mik.ua/orelly/perl4/lperl/index.htm)

* What is a regular expression and how can it be used?
* How can we use them in Perl?
* Write an expression that enable to find your name in a file regardless of any other content.

**During class**

* Examples of regular expressions.
* Examples of flags.
* Parsing files with regular expressions.
* **Implement (reg expressions.pl , perlexercise1 regx.pl)**

**Lab 2**

After this lab you should be able to use regular expressions in perl.

1. Using as an entry the golden beetle from Edgar Allan Poe, write a set of scripts that do the following using regular expressions.
   1. Count the number of times each letter e appears in the text.
   2. Show every word with more than 10 characters.
   3. Count the number of sentences the begin with’;’.
   4. Show the quoted “” segments in the story.
   5. Show the words that start with ‘g’ and end with ‘s’.
2. Using the file log.txt which contains access information to labcq-master.qro.itesm.mx. Write a script in Perl that, using regular expressions, shows the IP address and the date of the computers used to access web pages located in the lab-cq server.

**Reading 3**

Read and make notes of the content in [Learning Perl chapter 12 and 13](http://docstore.mik.ua/orelly/perl4/lperl/ch13_01.htm).

* Create safe and secluded folder where you will be testing these exercises.
* Create a couple of folders with some garbage files.
* Switch the files between the folders
* Delete those files which consume more space in your hard drive.

**During class**

* Compare different scripting languages, python, bash Java script, Perl, etc…
* **Implement (directory command from pearl.pl)**

**Lab 3 (extra activity for long semester)**

After doing this lab you should understand how to perform common sysadmin tasks (search and move files around folders, and write maintenance scripts).

Generate the copy files script, which checks in the file the keywords.txt the keywords that your script will search for. The program searches in several directories for files containing (in the name of the file or in the contents). The script should then write copies of the files into a new folder called “sensitive data”.

Generate the insert code script, which searches for the “copy script” you just made (wherever it is), and inserts instructions into the “copy script” that replace the keywords in the searched files with the phrase “all your bases are belong to us”.

Create the recovery script which searches all the files Generate a clean traces data, which checks in a .txt file the keywords that should be removed from the folder. The clean traces program searches in several directories for files containing the key words and deletes them from their folder and writes them into a new folder called “my traces”.