생명정보학에서 쓰이는
컴퓨터 Perl 언어의 기초 교육
Perl Programming (Basic syntax, Regular expression and file I/O)

2017/9/10
Jongsun Park, Ph. D.

성신여자대학교
SUNGSHIN UNIVERSITY

InfoBoss
Infomations & Systems

InfoBoss
Infomations & Systems

Plant Molecular Phylogeny Lab.

http://www.amborella.net/

Sungshin Women's University

Bio 40th Anniversary!!!

About Plant Molecular Phylogeny Lab.
People   Publications   Hot Issue
Korean Plant Family Identification

Herbarium (SWU)   Magnolia Genome Database
Plant Distribution Database
Links Press Release   Amborella.org
Specialist Certificate for the Botanical Field Work
EcoTrip

Lectures

공과대학 기초 학업능력 강화 지원프로그램: PERL programing   →   여기에 자료 있습니다~

Spring Semester

Plant Systematics

Plant Systematics Lab.

Science in Life (both semesters)

```perl
print "Hello, perl.\n";
```

```perl
my $a;
```

```perl
$a = 1;
```

```perl
$a = "ABC";
```

```perl
print "Hello perl!\n";
print $a;
```

```perl
print "a valie is ".$a."\n";
```

- Example of four fundamental arithmetic operation in perl:

PC01.pl

```
$a = 2 + 3;
$a = 2 − 3;
$a = 2 * 3;
$a = 2 / 3;
```

- Try to write the code (four codes for each calculation; PC01-1.pl to PC01-4.pl) for printing the result

  of four fundamental arithmetic operation.

Four fundamental arithmetic operations in Perl (2)   /Let's computer calculate!

InfoBoss
Infomations & Systems

- Complex fundamental arithmetic operations in perl:

PC02.pl

$a = 3 + 7 * 5 / 4;

$a = 3 + (7 * 5) / 4;

$a = (3 + 7) * 5 / 4;

- Try to write the code (three codes for each calculation; PC02-1.pl to PC02-4.pl) for printing the result

  of four fundamental arithmetic operation.

- Make numeric value as integer (e.g., 3141592 -> 3)

    PC03.pl

    ```
    $a = 3.141592;
    $b = int($a)
    ```

- Please write the code for converting 3.141592 to 3.1

    PC04.pl

- Let's get remainder with special operand from division equation.

    ```
    $a = (3 % 4);
    print $a."\n";
    ```

- Concatenation of strings: Please write the code for concatenating three strings ($a, $b, and $c) into
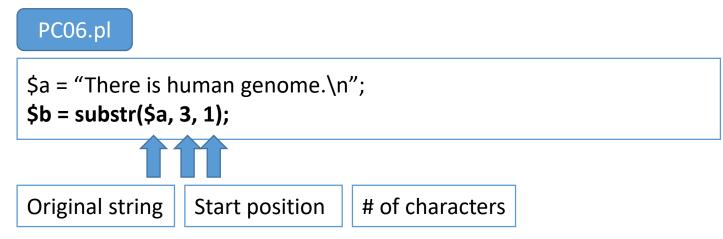
  one variable ($d).

  PC05.pl

  ```
  $a = "There is one apple";
  $b = " of which color is";
  $c = " red";
  ```
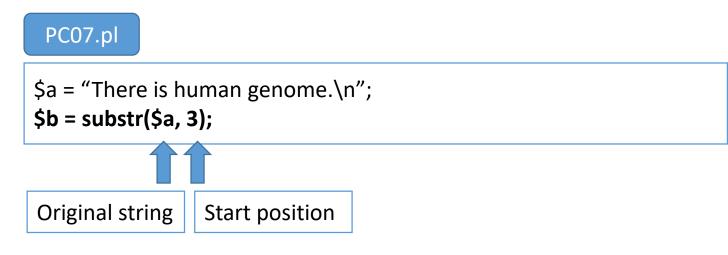
  ```
  $e = " blue";
  $f = " green";
  ```

- Make additional code for merging string $e and $f instead of $c.

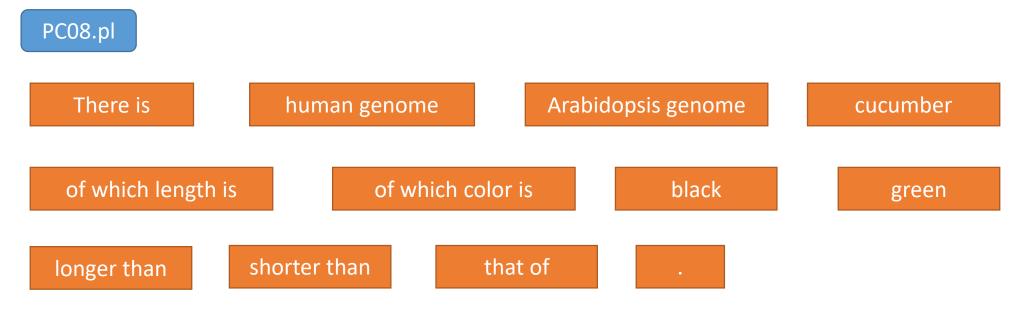- Extracting partial string from one variable: **substr** function.

PC06.pl

```
$a = "There is human genome.\n";
$b = substr($a, 3, 1);
```

Original string	Start position	# of characters

- substr can have only two parameters, third one can be omitted.

PC07.pl

```
$a = "There is human genome.\n";
$b = substr($a, 3);
```

Original string	Start position

- Complex code for controlling strings like below:

PC08.pl

There is	human genome	Arabidopsis genome	cucumber

of which length is	of which color is	black	green

longer than	shorter than	that of	.

- Write the program for printing following sentences

    - There is human genome of which length is longer than that of Arabidopsis genome.

    - There is cucumber of which color is green.

    - There is cucumber genome of which length is longer than that of Arabidopsis genome.

- Checking length of string: length

PC09.pl

```
my $seq = "ATGCTAGCTAGCTAGC";
print "Length : ".length($seq)."\n";
```

- Result from length can be stored into the variable.

PC10.pl

```
my $seq = "ATGCTAGCTAGCTAGC";
$c = length($seq);
```
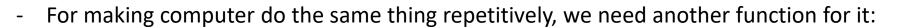
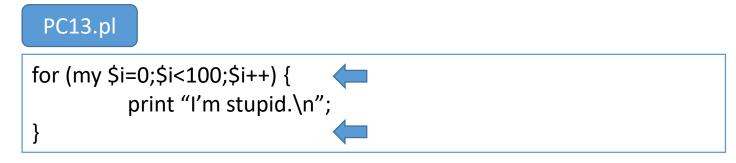- Make the program for getting lengths of two strings and print total length of the two string.

PC11.pl

$a = "There is good thing in the world.\n";

$b = "That is genome sequence!\n";

- Make the program for extracting 'good thing' from $a and print its length.

PC12.pl

- For making computer do the same thing repetitively, we need another function for it:

PC13.pl

```
for (my $i=0;$i<100;$i++) {          ⬅
        print "I'm stupid.\n";
}                                    ⬅
```

```
for ([initialize variable];[terminal condition];[operation]) {
        ~~

}
```

- 'condition' returns true or false based on the condition.

    - True case: 2 > 1

    - False case: 2 < 1

    - Once variables are added in the condition, the results will be **changeable** based on the value in

        variable.

- 'operation' part is usually increasing value in the variable.

- Let's combine other commands: Please write the code for calculating sum of numbers from **1** to **100**.

  PC14.pl

- Please write the code for making string variable which has "I'm smart." 40 times.

  PC15.pl

- Please write the code for calculating following equation:

$$a = \sum_{i=2}^{200} (i-1)*i$$

  PC16.pl

- Another simple loop command: while

```
my $i = 0;
while ($i < 10) {
        print "i = ".$i."\n";
        $i++;
}
```

```
while ([condition]) {
…
}
```

- If condition returns true, '…' will run.

- If not, loop will be done.

- Please write the code for calculating sum of numbers from **1** to **100** with while command.

    PC14-1.pl

- Please write the code for making string variable which has "I'm smart." 40 times with while

    command.

    PC15-1.pl

- Please write the code for calculating following equation with while command:

    $$a = \sum_{i=2}^{200} (i-1)*i$$

    PC16-1.pl

- 'If' phrase is *conditional statement* based on condition.

```
if (you are smart) {
        print "Yes, you are smart.\n";
} else {
        print "I'm sorry that you are not so smart.\n";
}
```

- 'you are smart' part is condition which returns true or false.

- 'else' means that if the condition is false.

PC17.pl

```
my $i = 3;
If ($i > 5) {
        print "i > 5.\n";
} else {
        print "i <= 5.\n";
}
```

- >, <, >=, <=, and != are possible to compare numeric values.

- Equation can be utilized in the condition, too.

- Special operands for comparing string: **eq** and **ne**.

PC18.pl

```
my $a = "I'm smart";
If ($a eq "I'm smart") {
        print "See, I'm smart :)\n";
} else {
        print "Oh, I'm not smart..\n";
}
```

- Try to use eq or ne for numeric values, what happened?

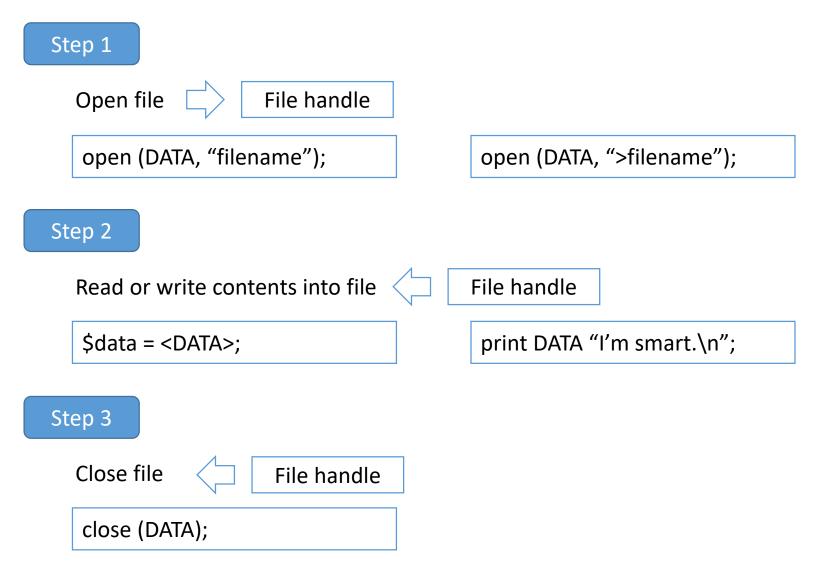- Please make the program to calculate following conditions:

    PC19.pl

    - From I = 1 to 100

    - If i is even, sum(i * 2)

    - If I is odd, sum (I * 3)

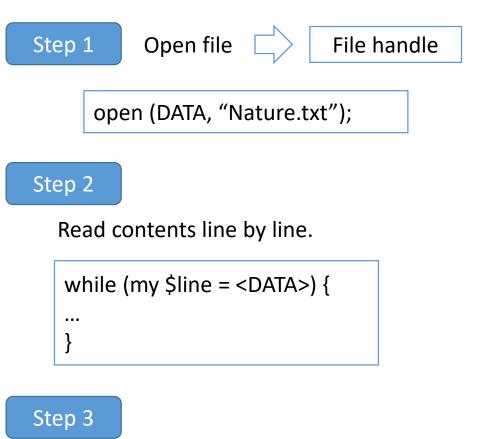   - Print final value.

- File operation in perl (and any other languages) consists of three steps:

Step 1

Open file ⇨ File handle

open (DATA, "filename");

open (DATA, ">filename");

Step 2

Read or write contents into file ⇦ File handle

$data = <DATA>;

print DATA "I'm smart.\n";

Step 3

Close file ⇦ File handle

close (DATA);

- Open text file (Nature.txt) and read content **line by line**.

Step 1   Open file ⟹ File handle

```
open (DATA, "Nature.txt");
```

Step 2

Read contents line by line.

```
while (my $line = <DATA>) {
…
}
```

Step 3

Close file

```
close (DATA);
```

- Write the program for counting total length of strings and total line from Nature.txt

PC20.pl

- Write the program for printing number of 'sequence' string in Nature.txt

PC21.pl

Regular expression is strange but efficient world!

- Regular expression was developed for finding patterns from string.

- Using it, not only finding patterns but also extracting values from string efficiently.

PC22.pl

```
my $line = "Red apple is delicious!";

if ($line =~ /apple/) {

        print "apple exists in the sentence\n";

} else {

        print "apple does not exist in the sentence\n";

}
```

- / is deliminator for regular expression.

- [ ] means character sets: **[A-Z]** means any characters from A to Z.

    - [0-9], [a-z], …?

- \w : alphabet characters

- \d : numerical characters

- + : repetitive characters

- Write the program with those regular expressions

  PC23.pl

- /\d/

- /\d+/

- \/w\

- /\w+/

- Write the program for checking whether file contains DNA or not (from Nature.txt)

  PC24.pl

- Let's extract value from string with regular expression.

PC25.pl

```
my $line = "7 red apple is delicious!";
if ($line =~ /(\d+)/) {
          print "extracted string : ".$1."\n";
}
```

- Parenthesis is special character for defining extract range of string based on pattern.

- $1 is special variable to keep extracted value and local variable.

- Write the code for extracting **number+one character** from the following sentences.

  - '13day' means thirteen days.

  - The string, ac3fd7e, is the serial number of product.

PC26.pl

Thank you for your attention!

If you have question,
please ask! =)

starflr@infoboss.co.kr