# Agenda

**Morning**

**Practice:** 30 minutes
**Environment Set Up:** 30 minutes
**Review Regular Expressions:** 45 minutes
**First Lab:** *Bob, the Teenager*

**Afternoon**

**Environment Set Up:** 30 minutes
**Looping and database access:** 1-1/2 hour
**Second Lab:** *SQL2CSV*

# Practice

Day 17 or Day 2
depending on how you look at it

# Cygwin

Open a Cygwin terminal

Change directory to your `$USERPROFILE`

Change directory to you perl development directory

Create a directory named `day-27` and confirm you made it by listing stuff

Remove the directory named `day-27`

Create a directory named `day-2`

## Cygwin

```
cd $USERPROFILE
cd dev/perl
mkdir day-27
ls
rmdir day-27
mkdir day-2
```

# Perl - Print Your Information

In the `day-2` directory, create `print-my-info.pl`, and open the file in your text editor.

Put in the standard header.

Declare a variable named `$me` and put in the following keys and their values:
a string “name”, a number “number_of_fingers”, and a list “friends_names”

Print the value of `$me` using `Data::Dumper`

Print just the first name from your list of friends

## Perl
## Print Your Information

```perl
#!/usr/bin/env perl
use warnings;
use strict;
use Data::Dumper;

my %me = (
  "name" => "Curtis",
  "number_of_fingers" => 10,
  "friends_names" => [
    "Stacie",
    "Bryan",
    "Heather"
  ]
);

print(Dumper(%me));
print("My first friend is ",
      $me{"friends_names"}->[0],
      ".\n");
```

## Environment Set-Up

## Install Perl Module
## `Test::Simple`

```
perl -MCPAN -e 'install Test::Simple'
```

## Regular Expressions

# Regex Matchers

| | Matches |
|---|---|
| `\s` | Whitespace such as spaces and tabs |
| `\w` | Word characters like letters and numbers |
| `\d` | Digits |
| `\S, \W, \D` | NOT whitespace, word chars, or digits, respectively |
| `[A-Z]` | Matches all the letters from A to Z |
| `[0-9]` | Matches the numbers 0 through 9, equivalent to `\d` |
| `[A-Z0-9_]` | Matches A to Z, 0 through 9, or the underscore |
| `foobar` | Matches the exact text “foobar” |
| `.` | Any one character |

# Regex Modifiers

| | **Matches** |
|---|---|
| ? | Zero or one instances |
| * | Zero or more instances |
| + | One or more instances |
| {n} | Exactly n instances |
| {m,n} | Between m and n instances, inclusive |
| {m,} | At least m instances |
| [^A-Z] | The caret inside the [] matches anything BUT |
| ^*regex* | The caret at the start anchors it to the beginning of a line |
| *regex*$ | The dollar sign at the end anchors it to the end of the line |

# Matching in Perl

```
print "Stop yelling." if ($text =~ m/^[A-Z\s]*!$/)
```

# Bob, the Teenager

Download from https://git.io/v1B5U

Unzip Bob.pm and Bob.t to your day-2 development folder

Open Bob.pm and Bob.t in your text editor

Run “perl Bob.t” on the command line

## Environment Set-Up

## Install Perl Modules
## `DBI` and `DBD::ODBC`

```
perl -MCPAN -e 'install DBI'
perl -MCPAN -e 'install DBD::ODBC'
```

## Create an ODBC Connection

## Accessing SQL Server

```
DBI->connect('dbi:ODBC:DSN=AdventureWorks');
```

# SQL2CSV

From the AdventureWorks database, get your data from the `Person.BusinessEntityContact` table with appropriate `JOIN`s to `Person.Person` and `Person.ContactType`

Your program should `die` if cannot connect to SQL Server

Your program should output a file named `Contact.csv`

Each row in the CSV must include the BusinessEntityID, the FirstName, the LastName, the Email, and the human-readable type of contact for each contact