PERL

Members:
Eguia, John Paul
Espantaleon, Jhosie Charles
Espra, Elram
Gamboa, Gilmar Dave P.

Coverage

- History
- Domain and Paradigm
- Features
- Demo

HISTORY OF PERL

1987

First version finally being released at the end of year 1987.

1988

Perl 2 was published.

1989

Perl 3 was released. Held support for binary data systems.

1991

Programming Perl book was released. Perl 4 was also released.

1994

Perl 5 was released. This release was almost a complete rewrite of the initial interpreter and included a large number of new features within the programming code.

1995

Release of the Comprehensive Perl Archive Network or CPAN.

1997

The next update of Perl 5 is released and contained a UNIVERSAL package.

2000

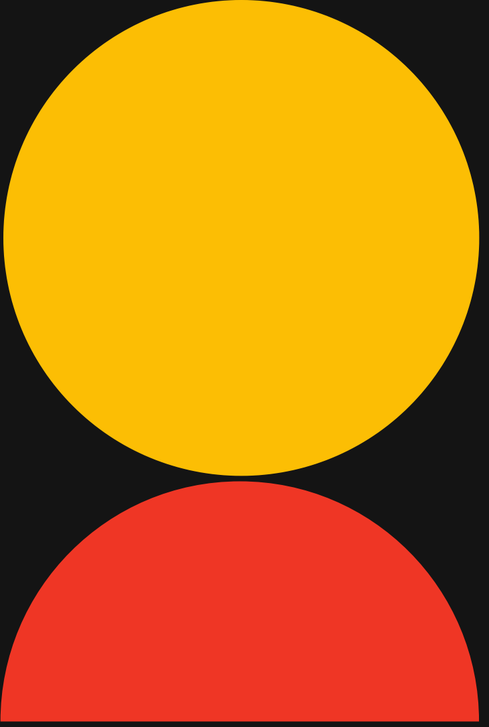
The release of Perl 5.6. It contains some significant changes such as 64-bit support, unicodes, etc.

1998

Perl 5.0005 was made public.

2001 onwards

Perl 6 still being developed. In 2013, Perl 5.8 version is most used. And is the most popular.



Domains and Paradigms

Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.

Perl is a declarative programming language, it is done with expressions or declarations instead of statements.

Perl is not a functional language in the sense that it also uses several other programming paradigms.

Procedural Programming

Object-Oriented
Programming

Procedural Programming

For example, to develop a simple Bank Account App procedurally:

```
let accounts = [];

function account(name, balance = 300){
  accounts.push({
    name: name,
    balance:  balance
  });
}

function getAccount(name){
  for(let i = 0; i < accounts.length; i ++){
    if(accounts[i].name === name){
      return accounts[i];
    }
  }
}

function deposit(name, amount){
  let account = getAccount(name);
  account.balance = account.balance + amount;
}

function withdraw(name, amount){
  let account = getAccount(name);
  account.balance = account.balance - amount;
}

function transfer(payer, beneficiary, payment){
  let payerAccount = getAccount(payer);
  withdraw(payerAccount.name, payment);
  let beneficiaryAccount = getAccount(beneficiary);
  deposit(beneficiaryAccount.name, payment);
}
```

Object Oriented Programming (OOP) is the use of self-contained code (objects) to develop applications.

```
class BankAccount{
  constructor(name){
    this.name = name;
    this.balance = 300;
  }

  deposit(amount){
    this.balance += amount
  }

  withdraw(amount){
    this.balance -= amount
  }

  transfer(beneficiary, payment){
    let payer = this;
    payer.withdraw(payment);
    beneficiary.deposit(payment);
  }
}
```

FEATURES

•Derives broadly from C. Perl is procedural in nature, with variables, expressions, assignment statements, brace-delimited blocks, control structures, and subroutines.

•Perl also takes features from shell programming. All variables are marked with leading sigils

•Perl takes lists from Lisp, hashes ("associative arrays") from AWK, and regular expressions from sed.

•Perl 5 added features that support complex data structures, first-class functions (that is, closures as values), and an object-oriented programming model

•All versions of Perl do automatic data-typing and automatic memory management.

DATA TYPES

SCALAR

Scalars are simple variables. They are preceded by a dollar sign ($). A scalar is either a number, a string, or a reference. A reference is actually an address of a variable, which we will see in the upcoming chapters.

Example:

```
$age = 25;              # An integer assignment
$name = "John Paul";    # A string
$salary = 1445.50;      # A floating point
```

ARRAY

Arrays are ordered lists of scalars that you access with a numeric index, which starts with 0. They are preceded by an "at" sign (@).

Example:

```
@ages = (25, 30, 40);
@names = ("John Paul", "Lisa", "Kumar");
```

HASHES

Hashes are unordered sets of key/value pairs that you access using the keys as subscripts. They are preceded by a percent sign (%).

Example:

```
%data = ('John Paul', 45, 'Lisa', 30, 'Kumar', 40);
```

OPERATORS

- Basic Arithmetic Operators (+, -, *, /, %, **)
- Assignment Operators (=, +=, -=, *=, /=, %=, **=)
- Auto-increment and Auto-decrement Operators (++, —)
- Logical Operators (&&, and, ||, or, not, !)
- Comparison operators (==, eq, !=, ne, >, gt, <, lt, >=, ge, <=, le)
- Bitwise Operators (&, |, ^, ~, <<, >>)
- Quote and Quote-like Operators (q{ }, qq{ })

DATA STRUCTURES

Perl has just two ways of organizing data:

- as ordered lists stored in arrays and accessed by position
- as unordered key/value pairs stored in hashes and accessed by name.

CONTROL STRUCTURE

- Perl is an iterative language in which control flows from the first statement in the program to the last statement unless something interrupts.

- if statement
- if/else statement
- if/elseif/else statement
- while statement
- until statement
- for statement
- foreach statement
- last operator
- next operator
- redo operator

DEMO