

Todo lo que puedes aprender
sobre Perl 6 en una noche sin
sueño



Germán Martínez Maldonado

Granada Perl Workshop 2018
Love4Perl

¿Quién soy yo?



¿Quién soy yo?



¿Quién soy yo?



- Defensor del Software Libre y los Derechos Digitales

¿Quién soy yo?



- Defensor del Software Libre y los Derechos Digitales
- Máster en Ingeniería Informática

¿Quién soy yo?



- Defensor del Software Libre y los Derechos Digitales
- Máster en Ingeniería Informática
- Trabajo actual: ----

¿Quién soy yo?



- Defensor del Software Libre y los Derechos Digitales
- Máster en Ingeniería Informática
- Trabajo actual: ----
- Coordinador en Interferencias

¿Quién soy yo?



- Defensor del Software Libre y los Derechos Digitales
- Máster en Ingeniería Informática
- Trabajo actual: ----
- Coordinador en Interferencias
- (Paladín del Software Libre a las órdenes de la Honorable Fundadora de Interferencias)

¿Quién soy yo?



- Defensor del Software Libre y los Derechos Digitales
- Máster en Ingeniería Informática
- Trabajo actual: ----
- Coordinador en Interferencias
- (Paladín del Software Libre a las órdenes de la Honorable Fundadora de Interferencias)

- Persona con más interés sobre la tecnología que tiempo tiene para aprender sobre ella

¿Se puede aprender Perl 6 en una noche?

¿Se puede aprender Perl 6 en una noche?

- SPOILER ALERT:

¿Se puede aprender Perl 6 en una noche?

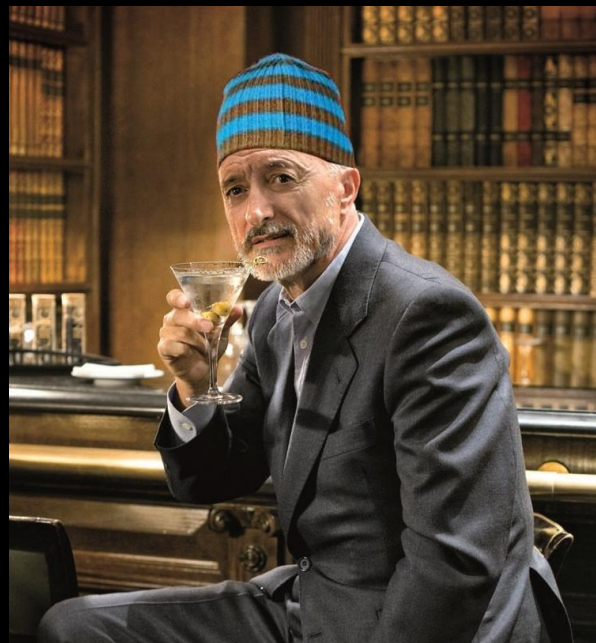
- SPOILER ALERT:
 - `say "no";`

¿Se puede aprender Perl 6 en una noche?

- SPOILER ALERT:
 - `say "no";`
- Pero habrá que intentarlo.

¿Se puede aprender Perl 6 en una noche?

- SPOILER ALERT:
 - say “no”;
- Pero habrá que intentarlo.



¿Qué es Perl?



¿Qué es Perl?

- Un lenguaje de programación diseñado por Larry Wall en 1987.

¿Qué es Perl?

- Un lenguaje de programación diseñado por Larry Wall en 1987.



¿Qué es Perl?

- Un lenguaje de programación diseñado por Larry Wall en 1987.
- Este tío es un genio.



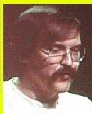
Larry Wall's *Very Own* Home Page



Howdy, world! This website is under construction.



(Did you ever see a website that *wasn't* under construction?)



Yes, I'm afraid chartreuse *is* one of my favorite colors...



No, I'm not going to change it just because it's not your favorite...

my **geek code** (3.1):

```
GC/CS/E/H/IT/L/M/MU/PA/P/S/SS/TW/0 d(++>)+ s: a>+>+>+>+ C++++$ UBAHS+>+>+ P++++(--)$ L !E? W+>+> N+++@ K++++>+>+>+>+>@ w$ !0 M->+ V--() P5+(-) PE(++ Y+ PGP->+ t+( ) 5 X? R>* tv@ b++>+> DI+++ D? G(-) e++>+>+@ h----() r+++ y++++  
\(explanation\)
```

my **ravings**

[1st](#) State of the Onion, 1997 Perl Conference.

[2nd](#) State of the Onion, 1998 Perl Conference.

[Perl, the first postmodern computer language](#), LinuxWorld Spring 1999.

[3rd](#) State of the Onion, 1999 Perl Conference.

I'm also keeping a [diary](#) all about my cornea transplant.

Perl

See my [Perl](#) page.

Web and CGI programming

I haven't done much of this. (Consider how lousy this web page is.) Consult the appropriate webpage.

rn

There are various derivatives being supported by various people. None of them are me.

patch

I recommend the version supplied by the GNU folks. <ftp://prep.ai.mit.edu/pub/gnu>.

metaconfig

No longer maintained. These days you're probably better off using something like GNU autoconf, one of the spiritual descendents of metaconfig.

my **publisher**

See [O'Reilly & Associates](#).

my **church**

¿Por qué Perl es importante?



¿Por qué Perl es importante?

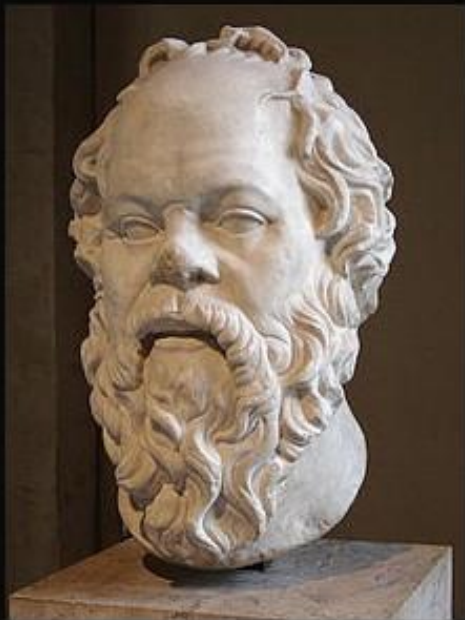
- Prácticamente sirve para todo.

¿Por qué Perl es importante?

- Prácticamente sirve para todo.
 - Tareas de administración en Unix

¿Por qué Perl es importante?

- Prácticamente sirve para todo.
 - Tareas de administración en Unix
 - Procesamiento de texto



There is only one good, knowledge, and one evil,
legacy code.

(Socrates)

¿Qué hace diferente a Perl 6?

“En Perl 6, decidimos que sería mejor arreglar el lenguaje que arreglar al usuario.”

Larry Wall

¿Qué hace diferente a Perl 6?

“Las cosas sencillas deberían continuar siendo sencillas, las cosas difíciles deberían ser más fáciles, y las cosas imposibles deberían ser difíciles.”

Larry Wall

Historia



Historia

- 19 julio 2000: anuncio del proceso de diseño de Perl 6

Historia

- 19 julio 2000: anuncio del proceso de diseño de Perl 6
- Comienzan las peticiones de comentario o RFC

Historia

- Apocalipsis: clasificación de Larry Wall de los RFCs

Historia

- **Apocalipsis:** clasificación de Larry Wall de los RFCs
- **Sinopsis:** versión depurada final del Apocalipsis

Historia

- Apocalipsis: clasificación de Larry Wall de los RFCs
- Sinopsis: versión depurada final del Apocalipsis
- **Exégesis**: versión práctica de la Sinopsis

Historia

- 25 de diciembre de 2015:

Historia

- 25 de diciembre de 2015:

**PRIMERA VERSIÓN ESTABLE
DE RAKUDO PERL 6**



Implementación

Implementación

- Rakudo Perl 6

Implementación

- Rakudo Perl 6
- No oficial

Implementación

- Rakudo Perl 6
- No oficial
- “Perl 6 será cualquier cosa que pase el conjunto de test oficial”

Rakudo Perl 6

Rakudo Perl 6

- Compilador para MoarVM y JVM

Rakudo Perl 6

- Compilador para MoarVM y JVM
- Escrito en C y Perl 6

Rakudo Perl 6

- Compilador para MoarVM y JVM
- Escrito en C y Perl 6
- Recomendada y multiplataforma

Rakudo Perl 6

- Compilador para MoarVM y JVM
- Escrito en C y Perl 6
- Recomendada y multiplataforma
- Nombre:
 - "Rakuda-dō" ('o' larga; 駱駝道): camino del camello
 - "Rakudo" ('o' breve; 樂土): paraíso

Instalar Rakudo Perl 6

Instalar Rakudo Perl 6

- Rakudo Star

Instalar Rakudo Perl 6

- Rakudo Star
- Rakudo Brew

Instalar Rakudo Perl 6

- Rakudo Star
- Rakudo Brew
- Rakudo Perl 6

¿Qué es Perl 6?

¿Qué es Perl 6?

- Lenguaje modular de propósito general

¿Qué es Perl 6?

- Lenguaje modular de propósito general
- Tipado gradual

¿Qué es Perl 6?

- Lenguaje modular de propósito general
- Tipado gradual
- Multiparadigma:

¿Qué es Perl 6?

- Lenguaje modular de propósito general
- Tipado gradual
- Multiparadigma:
 - Procedimental

¿Qué es Perl 6?

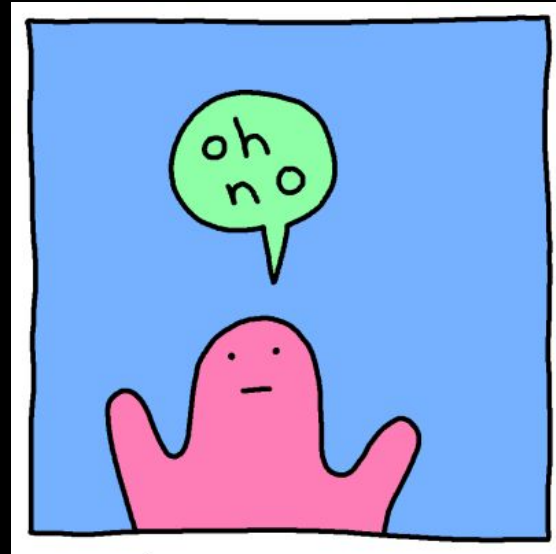
- Lenguaje modular de propósito general
- Tipado gradual
- Multiparadigma:
 - Procedimental
 - Orientado a objetos

¿Qué es Perl 6?

- Lenguaje modular de propósito general
- Tipado gradual
- Multiparadigma:
 - Procedimental
 - Orientado a objetos
 - Funcional

¿Qué es Perl 6?

- Lenguaje modular de propósito general
- Tipado gradual
- Multiparadigma:
 - Procedimental
 - Orientado a objetos
 - Funcional



Módulos

Módulos

- Sin soporte en CPAN

Módulos

- Sin soporte en CPAN
- Prototipo de sistema de módulos


Módulos

- Sin soporte en CPAN
- Prototipo de sistema de módulos
- Panda

Módulos

Perl 6 Modules

[ALL](#)[API 6](#)[ARCHIVE 3](#)[ASYNC 7](#)[AUDIO 13](#)[AUTHORISATION 3](#)[C 3](#)[CACHE 3](#)[CLI 12](#)[CLIENT 4](#)[COMMAND LINE 5](#)[COMPRESSION 5](#)[CONFIG 7](#)[CONFIGURATION 4](#)[CPAN5 38](#)[DATA 4](#)[DATABASE 5](#)[DEVEL 13](#)[ECOSYSTEM 3](#)[ECOSYSTEMS 3](#)[FASTCGI 3](#)[FCGI 3](#)[FILESYSTEM 7](#)[FONT 4](#)[FORMAT 3](#)[GRAMMAR 3](#)[HASH 4](#)[HTML 3](#)[HTTP 15](#)[IO 7](#)[JSON 9](#)[LINUX 6](#)[LOG 6](#)[LOGGER 4](#)[LOGGING 4](#)[MATH 6](#)[MESSAGING 3](#)[MUSIC 8](#)[NATIVE 5](#)[NATIVECALL 13](#)[NET 10](#)[NUMBER 3](#)[OBJECT 9](#)[PACKAGE-MANAGER 3](#)[PARSER 7](#)[PARSING 5](#)[PDF 8](#)[PERL5 27](#)[SEARCH 3](#)[SERIALIZATION 7](#)[SERVER 3](#)[SLANG 3](#)[SYSTEM 4](#)[TABLE 3](#)[TEMPLATE 4](#)[TERMINAL 12](#)[TESTING 9](#)[TEXT 15](#)[TIME 3](#)[UNIX 8](#)[USERS 4](#)[UTILS 46](#)[WEB 20](#)[WRAPPER 3](#)[XML 4](#)[YAML 5](#)[VIEW ALL TAGS](#)

 Use `from:source` to limit your search to a particular dist source, e.g. [from:gitlab](#) or [Test from:github](#)

Tipado gradual

Tipado gradual

- Dinámico:
 - my \$i = 43

Tipado gradual

- Dinámico:
 - `my $i = 43`
- Estático
 - `my Int $i = 43`

Orientación a objetos

Orientación a objetos

```
class Punto {  
  has $!x;  
  has $.y is rw;  
}
```

```
my $punto = Punto.new(x => 4.3);
```

```
$punto.y = 2;
```

Roles

Roles

```
class Mamífero is Vertebrado { ... }
```

```
class Perro is Mamífero { ... }
```

```
role Mascota { ... }
```

```
role Silvestre { ... }
```

```
role Perro_guía { ... }
```

Roles

```
class PerroGuía is Perro does Perro_guía {  
  ...  
}
```

```
my $perro = new Perro;  
$perro does Perro_guía;
```

Roles

```
class Humano {  
  has Perro $perro;  
}
```

```
role Ciego {  
  has Perro_guia $guia;  
}
```

Lista formal de parámetros

Lista formal de parámetros

```
sub hacer_algo(Str $cosa, Int $otra) {  
    // hace cosas  
}
```

Sigilos



Sigilos

- Invariantes, no dependen de la cantidad de elementos almacenados.

Sigilos

- Invariantes, no dependen de la cantidad de elementos almacenados.
 - \$: un solo elemento

Sigilos

- Invariantes, no dependen de la cantidad de elementos almacenados.
 - \$: un solo elemento
 - @: una secuencia de elementos

Listas, secuencias y arrays

Listas, secuencias y arrays

- **Iterable:** rol que gestiona conjuntos de datos que pueden recorrerse.

Listas, secuencias y arrays

- **Iterable:** rol que gestiona conjuntos de datos que pueden recorrerse.
- **List:** clase que hereda del rol iterable

Listas, secuencias y arrays

- **Iterable:** rol que gestiona conjuntos de datos que pueden recorrerse.
- **List:** clase que hereda del rol iterable
- **Sequences:** listas creadas con los elementos requeridos

Listas, secuencias y arrays

- **Iterable:** rol que gestiona conjuntos de datos que pueden recorrerse.
- **List:** clase que hereda del rol iterable
- **Sequences:** listas creadas con los elementos requeridos
- **Array:** lista de valores escalares

Elementos de un array

Elementos de un array

- Número de elementos de un array:
 - V5: \$#array+1 || scalar(@array)
 - V6: @array.elems

Elementos de un array

- Índice del último elemento de un array:
 - V5: \$#array
 - V6: @array.end

Elementos de un array

- Último elemento de un array:
 - V5: `$array[$#array]`
 - V6: `@array[@array.end]`

Evaluación perezosa

Evaluación perezosa

```
my @enteros = 0..Inf;
```

Evaluación perezosa

```
my @enteros = 0..Inf;
```

```
my @fibonacci = 1, 1, { $^a + $^b } ... *;
```

Simplificación sintáctica



Simplificación sintáctica

```
if es_verdadero() {  
    for @array {  
        ...  
    }  
}
```

Simplificación sintáctica

@array = 1, 2, 3, 4;

Simplificación sintáctica

Comparaciones

Comparaciones

```
if 10 <= $temperatura >= 40 {  
    say "Un día típico en Granada."  
}
```

Bloques



Bloques

- V5:

```
eval { # ... };  
if ($@) { warn "cuidao!"; }
```
- V6:

```
try { # ...  
    catch { warn "cuidao!" }  
}
```

Bloques

- V5:

```
eval { # ... };  
if ($@) { warn "cuidao!"; }
```


Bloques

- V5: `foreach (@cosas) { ... }`

Bloques

- V5: `foreach (@cosas) { ... }`
- V6: `for @cosas { ... }`

Bloques

- V5: `for ($i=0; $i<10; $i++) { ... }`

Bloques

- V5: `for ($i=0; $i<10; $i++) { ... }`
- V6: `loop ($i=0; $i<10; $i++) { ... }`

Expresiones regulares

Expresiones regulares

```
if (/REGEXP/) {  
    say "Is very difficult todo esto."  
}
```

Unicode



Unicode

- μ

Unicode

- μ
- ¾

Unicode

- μ
- ¾
- π

Unicode

- μ
- ¾
- π
- ✓

Unicode

- μ
- ¾
- π
- ✓
- ❄

Ejecución



Ejecución

- \$ perl6

Ejecución

- \$ perl6
- “strict” por defecto

Ejecución

- \$ perl6
- “strict” por defecto
- Mezclar Perl 5 y Perl 6

Conclusiones

Conclusiones

- Completo no, lo siguiente

Conclusiones

- Completo no, lo siguiente
- Simplificación y modularidad

Conclusiones

- Completo no, lo siguiente
- Simplificación y modularidad
- Escribe poco, haz mucho

FIN

