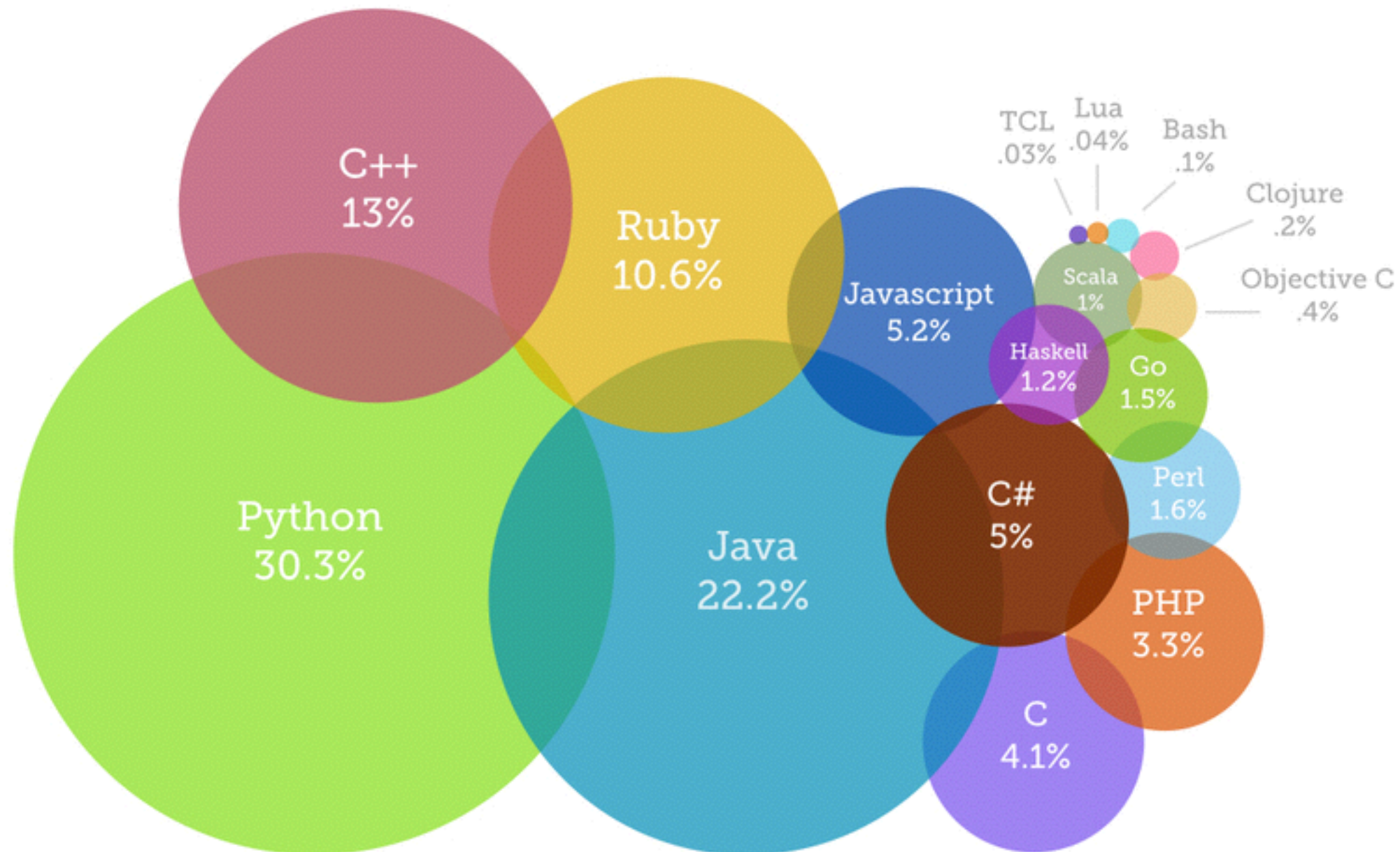




Lenguajes de Programación

Alejandro Cárdenas-Avendaño

Most Popular Coding Languages of 2014



Lenguajes

Compilados (Compiled)

- Deben ser compilados, es decir “traducidos” al código de máquina.
- Desarrollo más lento
 - Compilar -> Conectados -> Correr
- La sintaxis en general no es muy clara
- SON MÁS RÁPIDOS

Fortran, C, C++, Java

Guión (Scripted)

- El código es corrido por un interpretador línea a línea.
- El archivo de texto es el programa
- Desarrollo más rápido de escribir y experimentar

En línea

No compila Errores

Python, Javascript, Shell scripting.

C

- Hay que empezar todo desde cero, no trae nada.
- Se debe manejar la memoria
- Es muy portable
- No tiene Strings


C++

- Está orientado a objetos
- La sintaxis no es muy clara
- No es actualizado
- Las librerías pueden ser llamadas a otros lenguajes

Tiempos críticos

IDL

- ¡CUESTA!
- Fue impresionante en sus orígenes por las gráficas.
- Los intereses no están relacionados con ciencia como tal.
- Muchas librerías existen en IDL...

¿?

print, a

Python

- Es orientado a objetos
- Fácil de aprender, la sintaxis es “más natural”
- La memoria es manejada “internamente”
- El lenguaje ya tiene incluidas MUCHAS librerías y funcionalidades
- Está siendo continuamente actualizado
- Puede “citar” librerías escritas en C y C++

BATERIAS INCLUÍDAS

Editores de texto

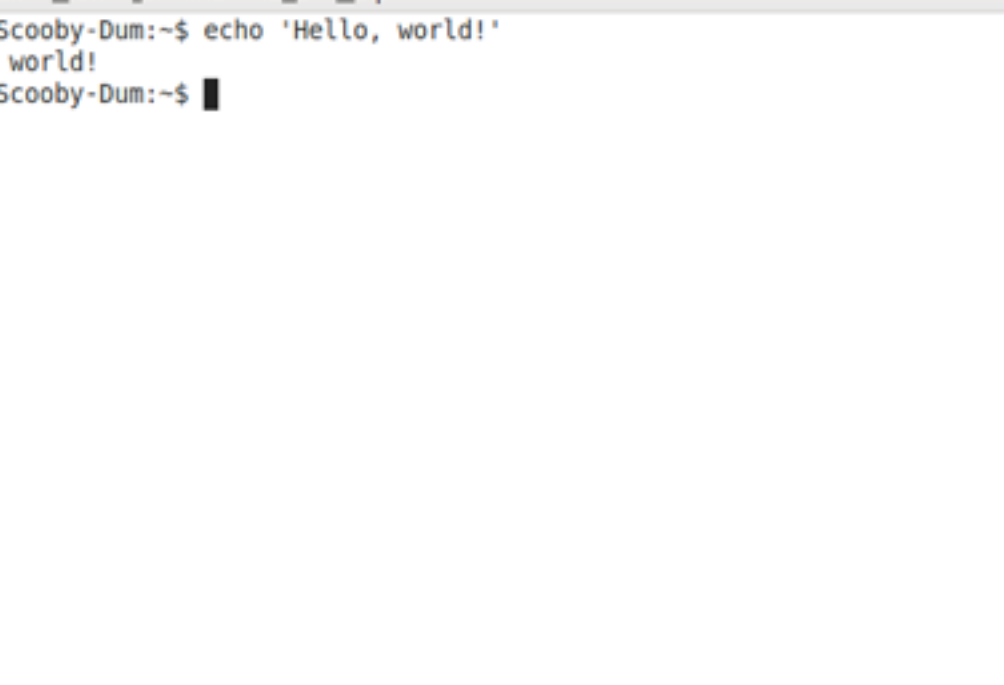
- La mayoría del tiempo se está trabajando en el.
 - Fácil de Usar
 - Ayudas
 - Numeración de Líneas
 - Conozca nombres prohibidos
 - Señale la sintaxis
- Son como una religión

Terminal

- Hay que personalizarlas

```
pilot:~/git/lufa-ftdi* ~/git/terminal/terminal.py --all /dev/ttyUSB*
/dev/ttyUSB0, 115200 baud
/dev/ttyUSB1, 115200 baud
/dev/ttyUSB2, 115200 baud
/dev/ttyUSB3, 115200 baud
/dev/ttyUSB4, 115200 baud
^C to exit
-----
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[button pressed!]
[got byte X][[got byte X]ot byte X][got byte X][[got bgo[t byte X]te X]got byte X][got byte X]
```

- Que no se pixele
- Que sea fácil de leer
- Que no sea aburrida



```
bruce@Scooby-Dum: ~  
File Edit View Terminal Tabs Help  
bruce@Scooby-Dum:~$ echo 'Hello, world!'  
Hello, world!  
bruce@Scooby-Dum:~$
```

[illegible]

Computación científica

- **FLOPS**: floating point operations; Fortran, C
- **Estadística**: MAT\$LAB, R, Numpy
- **Big Data**: SP\$SS
- **Simbólico**: Map\$le, Mathem\$atica,
- **TODO**: Python, Ruby, Scala, Haskell

Importante

Tiempo

¿Cuál? -> Depende

Minimizar

Complejidad

Escribir

Modificar

¿Cómo debería ser?

- Llegar a un computador que nunca antes se haya visto
- Digitar “un par” de comandos
- Obtener el resultado deseado
- ...

¿Qué debo esperar de un leguaje?

El proceso científico “Moderno”

- Aplicar a un grant
- Observar y Explorar fenómenos
- Generar un hipótesis
- Proponer modelos que expliquen el fenómeno
- Probar las predicciones del modelo
- Modificar la teoría y repetir todo el proceso
- Publicar

- Observar y Explorar fenómenos
- Permitir a cualquier persona manipular los datos fácilmente
- Muestre la información y permita hacer estadística básica
- Permita realizar variaciones en las visualizaciones
- Interactue con la comunidad

- Generar un hipótesis
- Aprendizaje no supervisado (Agrupamiento, redes neuronales)
- Reducción de dimensionalidad
- Compacto y que interactue con varios modelos

- Proponer modelos que expliquen el fenómeno
- Aprendizaje supervisado (Regresiones, clasificación)
- En genera una caja de herramientas
- Comunidad en desarrollo

- Probar las predicciones del modelo
- Pruebas automáticas (Significancias, Sensibilidad)
- Proponer pruebas
- Optimización
- Comparar

- Modificar la teoría y repetir todo el proceso
- ¡Notebook!
- Fácil de corregir (por mí y por la comunidad)
- Sesiones interactivas

- Publicar
- Gráficos “bonitos” y “agradables”
- Estadística apropiada
- “Verificables” (Que la matemática sea “uno a uno”)
- “Reproducibile” (código simple)
- “Repetible” (Código abierto, datos)

¿De qué estamos hablando?

- “La elegancia no es una opción” R. O’Keefe
- Una sintaxis:
 - Explícita
 - Concisa
 - Consistente
- Abstracciones “terrenales”
- Corra en múltiples núcleos
- Documentación
- Ejemplos
- Tutoriales

Referencias

- Muna, D & Price-Whelan, A. SciCoder Workshop.
scicoder.org
- Peter Norvig, “What to demand from a Scientific Computing Language”, Mathematical Sciences Research Institute, 2010.
- Eric Mjolsness, Dennis DeCoste, “Machine Learning for Science: State of the Art and Future Prospects”, **Science** 14 September 2001, Vol. 293 no. 5537 pp. 2051-2055.