

Introducción Language Python

Temario de la clase

- Características de python.
- Porque programar en python? Cuando si y cuando no?
- Porque los pythónicos son tan entusiastas?
- Python como una calculadora.

Que es Python?

Lenguaje de muy alto nivel (sintaxis comprensible y muy sencilla).

Lenguaje interprete (No necesita de compilación).

Lenguaje estructurado. La **tabulación** es parte de la sintaxis.

Lenguaje orientado a objetos.

Lenguaje utilizado en la mayoría de las aplicaciones desarrolladas por Google.

Youtube esta hecho en python.

Lenguaje utilizado por Industrial Light & Magic en la producción de Star Wars y por DreamWorks Animations.

Porque Python?

Programas muy compactos (3-4 veces mas corto que en fortran o C).

Programas legibles.

Lenguaje estructurado y orientado a objetos.

Es un lenguaje open-source (gnu).

Muy fácil /de integrar con/integrador de/ otros lenguajes C/Fortran/Java.

Enorme comunidad de desarrolladores y usuarios, también en el ámbito científico.

Programar disfrutando!

Cuando no usar Python?

Python es un lenguaje de muy alto nivel, eso trae aparejado ventajas y desventajas.

Entre las segundas encontramos:

La ejecución de fórmulas matemáticas intensivas (bucles de > 10000 ciclos) suele ser lenta. Para computación numérica de alta performance el mas eficiente es Fortran seguido de C.

El control de dispositivos de la computadora (hardware) es indirecto e insuficiente. El C es el lenguaje con el mayor control de dispositivos en forma eficiente (Los SOs estan en C!).

La estructura flexible puede ser peligrosa si no se siguen reglas. (Un mal uso puede llegar a bugs difíciles de rastrear).

Python en la física

Existe una enorme cantidad de librerías científicas en Python todas ellas disponibles como open-source.

Desde procesos numéricos básicos: ej. integración, matrices.

Librerías de graficación en 3D.

Librerías para el tratamiento estadístico de datos.

Librerías de inteligencia artificial.

Muchos de los programas que desarrollamos en el grupo de investigación son adaptaciones o utilizan librerías ya desarrolladas por otros grupos.

Para que usamos nosotros el python?

1. Todo tipo de prueba o test sencillo que se quiera realizar en python.
2. Desarrollo de métodos y algoritmos: se comienza en python y luego se lleva a fortran
3. Las simulaciones se realizan en fortran (cluster) pero luego el análisis lo realizamos en python, la graficación en python (o idl).
4. Desarrollo de software para empresa Claro. Manejo estadístico. Todo python.

La manera pythonica de programar

Reglas para programar en python. Comando

```
>>> import this
```

- Lindo es mejor que feo
- Explícito es mejor que implícito
- Simple es mejor que complejo.
- Directo es mejor que anidado.
- Esparso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para romper las reglas
- El pragmatismo por sobre la pureza
- Los errores deben siempre hacerse notar.
- Debería haver una y preferiblemente solo una forma obvia de hacerlo.
- Ahora es mejor que nunca.
- Si la implementación es difícil de explicar, entonces es una mala idea.

Como usar python

En una terminal shell/bash. Desde línea de comando:

```
$ python
```

Si se quiere ejecutar en forma remota (en otra computadora)

```
$ ssh -X usuario@computadora
```

```
$ ssh -X usuario@200.45.54.94
```

Luego se puede ejecutar el python:

```
$ python
```

Los archivos python tienen la extensión **.py**

Exportar un editor de textos (debe entender python para la tabulación).

Python como una calculadora

Operaciones aritméticas

```
>>> 55+15
```

```
>>> -33+12
```

```
>>> 5.2/3.1 + 2
```

```
>>> 4**2
```

```
>>> 4**0.5
```

Python como una calculadora. Tipo de variables.

```
>>> 5/2
```

```
>>> 5/2.0
```

```
>>> 3.0e3
```

```
>>> 'Hola'
```

Tipos de variables: Enteros. Flotantes. Cadena de caracteres. Lógicas.
Números complejos

```
>>> a=5+2j
```

```
>>> type(a)
```

```
>>> print a.real
```

```
>>> print a.imag
```

Transforma un número flotante/entero en complejo:

```
>>> complex(5.0)
```

Python como una calculadora. Orden de las operaciones.

Orden de las operaciones aritméticas.

```
>>> 5*3**2
```

```
>>> (5*3)**2
```

```
>>> 5+3**2
```

```
>>> (5+3)**2
```

1. **, 2. *, /, 3. +, -. Se altera con ()

Variables

Supongamos que queremos tener disponible el valor de π para usarlo en varias ocasiones, en lugar de tener que tipearlo cada vez que lo necesitemos, entonces asignamos a una variable el valor de pi:

```
>>> pi=3.14159
```

Luego la variable pi tiene guardado el valor 3.14159, compruebalo poniendo:

```
>>> pi
```

Si ahora queremos calcular el perímetro de una circunferencia de radio 20cm,

```
>>> 2*pi*20
```

Quizas también nos convenga guardar el radio en una variable:

```
>>> radio=20
```

la circunferencia es entonces:

```
>>> 2*pi*radio
```

y la superficie de la circunferencia es:

```
>>> radio*pi**2
```

Variables. Reasignación

Si durante la ejecución necesitamos reasignar el valor de la variable, supongamos ahora tenemos una circunferencia de 15.5cm, redefinimos la variable:

```
>>> radio =15.5  
>>> radio  
>>> 2*pi*radio  
>>> radio*pi**2
```

La variable ha tomado el nuevo valor, notar además que cambiamos el tipo de variable, al principio la definimos como entera y luego como flotante.

Salidas por pantalla: print

Para que la computadora ponga como salida algo que nosotros queremos mostrar se usa el comando print.

```
>>> print 'Hola'
>>> radio=2.02
>>> unidad_radio='cm'
>>> print 'El radio es de ',radio,unidad_radio
```

Si queremos imprimir varias variables y/o cadenas de caracteres las separamos por comas.

A las instrucciones que le damos a la computadora le llamamos comandos, print es un comando.

Primer programa python

La mayoría de las veces lo que tendremos son muchas líneas de código por lo que queremos guardarla para usarla a futuro. Para esto al código lo tenemos que guardar en un archivo que denominamos “programa”.

Editamos un archivo:

```
$ xemacs simple.py  
#!/usr/bin/env python  
pi=3.141592  
radio=20.0  
per=2*pi*radio  
sup=pi*radio**2  
print 'El perimetro es: ',perim  
print 'La superficie es: ',sup
```

Ejecución del primer programa python

Para ejecutar un programa python lo que hacemos en una terminal shell/bash es:

```
$ python simple.py
```

Si el programa se encuentra en otro directorio se le da el camino completo:

```
$ python /home/pulido/curso/programacion/pyt/simple.py
```

Si el programa lo han hecho ejecutable (con el

```
chmod +x simple.py
```

)

entonces:

```
$ ./simple.py
```


Comentarios en el programa

A menudo queremos comentar las líneas de programa para luego recordar lo que hicimos.

Para explicar lo que hacemos en el código a otro programador.

Para referenciar el objetivo, que es lo que hace, cuando lo hicimos, cuando lo modificamos, que cosas necesitamos agregarle, etc.

Los comentarios de una sola línea se hacen con #:

```
# (todo lo que sigue detrás del símbolo python lo interpretará como un comentario)
```

```
5+8 # suma
```

Comentarios de varias líneas se hacen con: """ (triple comillas)

```
""" Programa para el calculo del  
perimetro y la superficie de una circunferencia.
```

```
MP. [2016-08-10]
```

```
TODO. Agregar el volumen de una esfera
```

```
"""
```

```
pi=3.141592
```

```
radio=20.0
```

```
# Calcule del perimetro
```

Entrada de valores

Supongamos que queremos diseñar un programa para estudiantes de la primaria (que no saben programar) que midan el diámetro de distintos objetos circulares y el programa que les calcule el perímetro y la superficie. El programa les debería pedir que introduzcan el valor del diámetro.

El comando que detendrá la ejecución y pedirá para que el usuario ingrese una cantidad es el **input**, en general el valor que se ingrese se guardará en una variable:

```
d=input("introduzca el diametro del objeto: ")
```

Entrada de valores

Entonces el programita para los estudiantes de primaria con el **input** sería:

```
pi=3.141592
d=input("introduzca el diametro del objeto: ")
# Calculo del perimetro
per=pi*d
sup=pi*(d/2.)**2 # superficie
print 'El perimetro del objeto es: ',perim
print 'La superficie del objeto es: ',sup
```

A este programa le faltan chequeos para cuando el usuario introduce algo erroneamente.