

Python (VI): Módulos

Programación (tics 100)
Semestre 01/2021

¿Qué veremos hoy?

Módulos, o cómo me aprovecho del trabajo de los demás



Módulos

Hasta el momento hemos utilizado funciones básicas en nuestros programas:

```
print("texto ", var)
```

➤ función que recibe múltiples parámetros y lo despliega por pantalla, no retorna ningún valor.

```
int(var o valor)
```

➤ función que recibe un valor, y retorna el mismo valor transformado en un entero

```
input("texto")
```

➤ función que recibe un parámetro, lo despliega por pantalla, recibe un valor desde el teclado y retorna un texto.

Módulos

Las funciones utilizadas fueron previamente creadas por otras personas, y su código está definido en una biblioteca o **módulo**.

En el caso de estas funciones básicas, el módulo se llama **builtins** y Python se encarga de dejarlas disponibles para nosotros.

import

Sin embargo, existen muuuuuuuuuuuchos módulos en Python disponibles para ser usados por Ustedes en sus programas.

Para hacer uso de ellos, se debe importar el módulo deseado y luego usar la función correspondiente

Importando módulos

```
import <nombreMódulo>  
nombreMódulo.nombreFunción(<parametros>)
```

Módulos

Dentro de los módulos más útiles en Python podemos encontrar:

- `math` para operaciones matemáticas
- `random` para generar números aleatorios
- `matplotlib` para graficar
- `numpy` y `pandas` para análisis de datos

Entre otros....¡muchos otros más!

Un ejemplo: Módulo math

El módulo math nos permite ocupar operaciones matemáticas útiles.

Para usarlo, debemos importarlo. Esto se hace al principio de nuestro archivo

```
import math
```

¿Qué cosas tiene el módulo math?

Para obtener información de un módulo en particular, por ejemplo, las funciones existentes, vaya al terminal, importe el módulo y llame a la función: **dir(<nombre módulo>)**

Ejemplo:

```
import math  
dir(math)
```

```
['__doc__', '__file__', '__loader__', '__name__',  
'__package__', '__spec__', 'acos', 'acosh', 'asin',  
'asinh', 'atan', 'atan2', 'atanh', 'ceil',  
'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf',  
'erfc', 'exp', 'expm1', 'fabs', 'factorial',  
'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd',  
'hypot', 'inf', 'isclose', 'isfinite', 'isinf',  
'isnan', 'ldexp', 'lgamma', 'log', 'log10',  
'log1p', 'log2', 'modf', 'nan', 'pi', 'pow',
```


¿Y cómo se para qué sirven?

Para obtener información de una función (o módulo) en particular, por ejemplo, la función `math.sin`, vaya al terminal, importe el módulo y llame a la función: **`help(<nombre módulo>.<nombre función>)`**

Ejemplo:

```
import math
help(math.sin)
```

Help on built-in function sin in module math:

```
sin(x, /)
```

Return the sine of x (measured in radians).

Un ejemplo: Módulo math



Veamos algunas funciones importantes del módulo **math**

math.ceil(x): retorna el entero más pequeño que sea igual o mayor que x

math.exp(x): retorna el valor exponencial de x

math.fabs(x): retorna el valor absoluto de x

math.factorial(x): retorna el factorial de x

math.floor(x): retorna el entero más grande que sea igual o menor que x

math.log(x, baseLog): retorna el logaritmo de x en base baseLog

math.pow(x, y): retorna x elevado y

math.sqrt(x): retorna la raíz cuadrada de x

Ejercicio



Tiempo : 10 minutos

Cree un programa que le **solicite** al usuario un número z y **un número de iteraciones** $numIter$ por pantalla y calcule la sumatoria:

$$\sum_{k=0}^{numIter} \frac{z^k}{k!}$$

`math.factorial(x)` : retorna el factorial de x :

`math.pow(x,y)` : retorna x elevado y



importamos el módulo

```
import math
z = int(input("ingrese numero: "))
numIter = int(input("ingrese numero iteraciones: "))
suma = 0
for k in range(0,numIter+1):
    suma = suma + math.pow(z,k) / math.factorial(k)
print("el valor es ",suma)
```

Módulos time y random

Veamos algunas funciones importantes del módulo **time**

time.sleep(x) : el programa se pausa por x segundos

time.time() : devuelve el número de segundos desde el 1 de Enero de 1970
(año que se utiliza como valor inicial en Unix)

time.ctime(x) : convierte el tiempo expresado en segundos en el formato de la fecha actual: Día de la semana, Mes, día, hora, año.

Veamos algunas funciones importantes de la biblioteca **random**

random.randint : retorna un número aleatorio entero entre x e y.

random.random () : retorna un número aleatorio entre 0 y 1.

random.uniform (x,y) : retorna un número aleatorio uniforme entre x e y.

Ejercicio



Cree un programa que genere un número aleatorio del 1 a 10 y pida al usuario adivinarlo en 3 intentos o menos.

*Tiempo : 5
minutos*

```
import random

num1 = random.randint(1,10)

while acierto = False and intentos < 3:
    n = int(input('Ingresa un número'))
    if n == num1:
        acierto = True
    else:
        intentos = intentos + 1

if acierto:
    print('Adivinaste')
else:
    print('Perdiste')
```

random.randint (x,y):
retorna un número
aleatorio entero entre
x e y.

Ejercicio



Tiempo : 10
minutos

Cree un programa que genere una multiplicación entre dos números en forma aleatoria y le pida al usuario resolverla. Si el usuario se demoró más de 10 segundos en resolverla, despliegue el mensaje “eres muy lento, el resultado era X”. Caso contrario, verifique si el usuario pudo resolver la multiplicación en forma correcta, felicitándolo o mostrando la solución de la multiplicación.

```
import time
import random

time1 = time.time()
num1 = random.randint(1,100)
num2 = random.randint(1,100)
print("¿Cuánto es ", num1, " por ", num2, "? ", end="")
res = int(input())
time2 = time.time()

if (time2-time1>10):
    print("Eres muy lento, el resultado era ", num1*num2)
else:
    if (num1*num2==res):
        print("FELICITACIONES")
    else:
        print("Te equivocaste, el resultado era ", num1*num2)
```

time.time(): devuelve el número de segundos desde el 1 de Enero de 1970 (año que se utiliza como valor inicial en Unix)

random.randint (x,y): retorna un número aleatorio entero entre x e y.

Módulo sympy

Otro módulo interesante es **sympy**.
sympy permite hacer matemática simbólica

```
from sympy import *  
  
x = symbols('x')  
a = Integral(cos(x)*exp(x), x)  
print(Eq(a, a.doit()))
```

¿Se fijaron cómo importamos en este caso? El comando `from ... import ...` nos permite importar funciones de un módulo y usarlas en forma directa (sin usar `nombreMódulo.función`)

Eq(Integral(exp(x)*cos(x), x), exp(x)*sin(x)/2 + exp(x)*cos(x)/2)

$$\int e^x * \cos(x) = \frac{e^x * \sin(x)}{2} + \frac{e^x * \cos(x)}{2}$$

Profe... estoy usando Python en mi computador y me dice

**ModuleNotFoundError: No
module named 'sympy'**

pip

pip es un **sistema de gestión de paquetes** utilizado para instalar y administrar paquetes de software escritos en Python. **pip3** es la versión para Python 3.

Para usarlo, en el terminal, o consola, o command prompt, usen:

```
pip3 install sympy
```

Ejercicio



Tiempo : 5 minutos

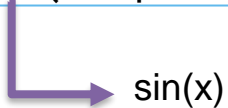
Para ver documentación de sympy, visita <https://docs.sympy.org/latest/tutorial/simplification.html> (o usen `dir` y `help`)

Luego escribe un programa que simplifique la expresión

$$1/x + (x*\sin(x) - 1)/x$$

```
from sympy import *

x = symbols('x')
a = 1/x + (x*sin(x) - 1)/x
print(simplify(a))
```



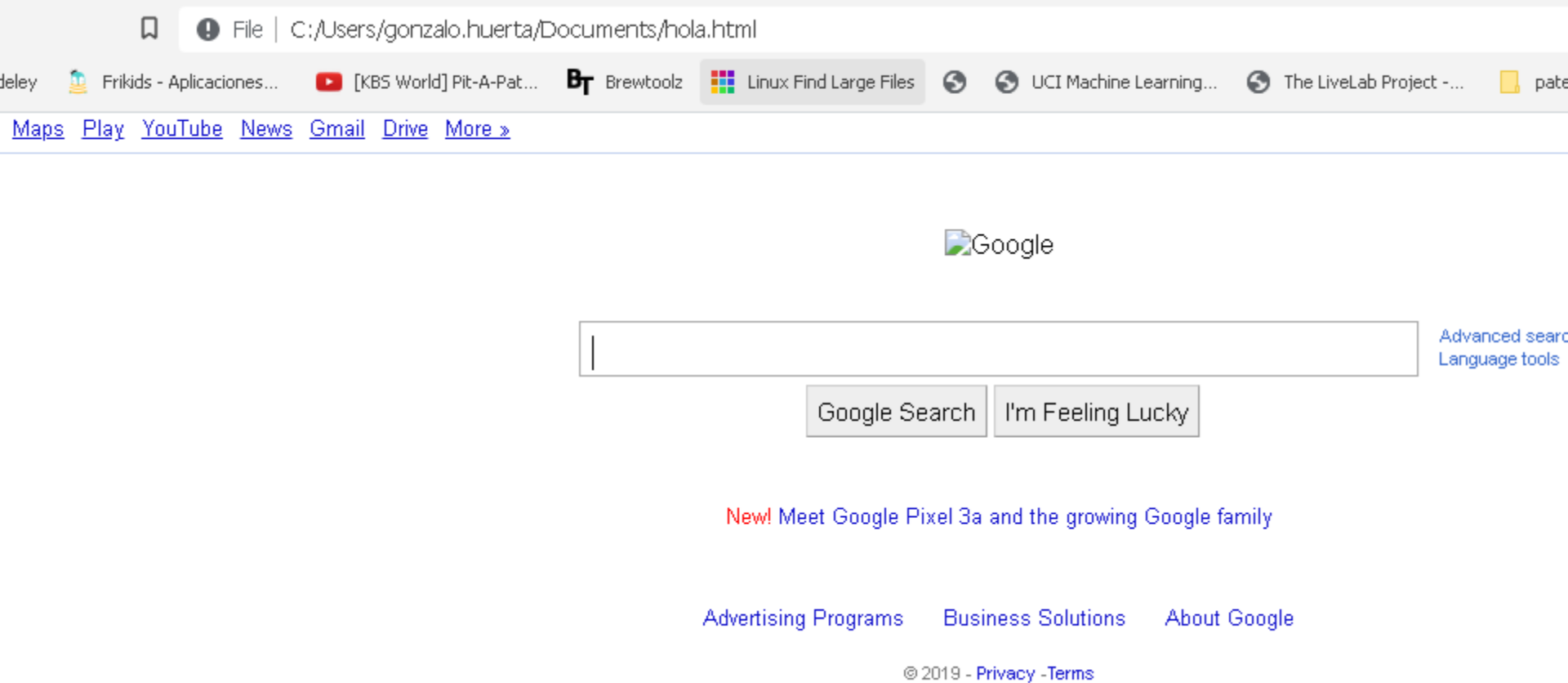
Módulo request

Un módulo que se usa bastante es **request**.
Lo pueden ver como un navegador web simple

```
import requests  
  
x = requests.get('http://www.google.com')  
print(x.text)
```

Copien el resultado y guárdenlo en un archivo llamado google.html
Ahora abran el archivo con Chrome u otro navegador...¿qué ven?

“Google”



Ejercicio



Tiempo : 5
minutos

Usa el módulo requests para obtener el contenido de una página que te indique el usuario...será el inicio de tu navegador 😊

```
import requests

pagina = input("Ingresa una página a navegar")
x = requests.get(pagina)
print(x.text)
```

¿Preguntas?

Actividad práctica

Desafío

- Aprende a usar más el módulo requests y mejora tu navegador.
- En grupos de 4
- Difícil que a todos se les ocurra la misma mejora :)
 - ▶ ¡sean creativos y háganlo Ustedes!
- Elijan un tema que les sea interesante y no complejo

Ejemplos

- ** Guardar la página como texto
- * Buscar una palabra escrita en la página
- *** Descargar las imágenes asociadas

¡Lo que ustedes quieran!