Ayudantía 5

Profesor: Carlos Alvarado **Ayudante: Pablo González**

Actividad	Objetivo
Repaso	• Funciones. • HTML. • Scraping.
Ejercicios con funciones	• Espejo • Busca números
Ejercicios de Scraping	• Extrayendo la etiqueta que queremos. • PIB por país.
Propuestos	• ¿Qué tienes para mi Kattis?
Cierre	• Dudas ayudantía

Repaso

Funciones

Las funciones nos sirven para llamar por un nombre a un conjunto de acciones determinadas que necesitaremos realizar en el futuro. Vea el siguiente ejemplo y pruebe agregar la linea "global s" al inicio de la función. ¿Cómo cambiarán los resultados?

In [2]:

```
s = 24

def mi_funcion(): #Definimos una función de la siguiente forma def nombre_de_mi_funcion():
    s = 2
    print('El valor de s es localmente %i' %s)
    return s

mi_funcion()
print('El valor de s es globalmente %i' %s)
```

```
El valor de s es localmente 2
El valor de s es globalmente 24
```

También podemos establecer parámetors default, de forma que no obtengamos algún error si es que alguien olvida definir una acción.

In [9]:

```
def pizarra(texto, num=50):
    print(texto * num)

pizarra("No volveré a olvidar escribir un parámetro ")
```

```
No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a
olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escr
ibir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un pará
metro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No vol
veré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvid
ar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir
un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro
```

```
No volvere a olvidar escribir un parametro No volvere a olvidar escribir un parametro No volvere a
olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escr
ibir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un pará
metro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No vol
veré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvid
ar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir
un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro
No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a
olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escr
ibir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un pará
metro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No vol
veré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvid
ar escribir un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir
un parámetro No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro
No volveré a olvidar escribir un parámetro No volveré a olvidar escribir un parámetro
```

Además, las funciones también soportan control de flujo dentro de ellas, de forma que podemos evaluar diferentes casos, como vieron en la tarea 2.

In [12]:

```
def menor_valor(a, b):
    if a < b:
        print(a, 'es menor')
    elif a == b:
        print('%i es igual a %i' % (a,b))
    else:
        print(b, 'es menor')

# directly pass literal values
v1, v2 = input("Por favor escriba 2 números: ").strip().split()

menor_valor(int(v1),int(v2))
```

```
Por favor escriba 2 números: 5 5
5 es igual a 5
```

HTML

Html no es un lenguaje de programación, sino uno de etiquetado. Este nos permite marcar a qué corresponde cada elemento de un sitio web, definir identificadores y/o clases para cada elemento, para luego añadir diseños o establecer acciones con otros programas. A continuación podrán ver una estructura básica de html.

Títulos:

Título 1

Título 2

Título 3

Espaciado:
Contenedor:
Párrafo:
Esto es un parrafo.
Link: Link

Existen muchas más etiquetas en html como veremos ahora. Dirígase a este sitio en donde continuaremos viendo lo básico de este lenguaje, utilizando la opción "inspeccionar" del navegador.

Web Scraping básico:

En ocasiones querremos extraer ciertos datos o información desde la web de forma automática para su posterior análisis. Esto se puede lograr mediante técnicas de web-scraping que nos permiten extraer específicamente los elementos que queremos de la web. Para ello, utilizaremos las librerías "requests" y "bs4" (ambos incluidos en Anaconda), además del conocimiento básico que tienen de HTML.

In [13]:

```
import requests
from bs4 import BeautifulSoup
page = requests.get("http://economiayadministracion.uc.cl/category/novedades/")
page #Debiesemos recibir un <Response [200]>, indicando que obtuvimos correctamente la página web.

#print(page.content) #Claramente necesitaremos limpiar este código.
```

Out[13]:

```
<Response [200]>
```

Para poder limpiar toda la información que obtuvimos y posteriormente poder navegar a través de esta, utilizamos la módulo BeautifulSoup de la librería bs4 mencionada anteriormente.

In [15]:

```
soup = BeautifulSoup(page.content, 'html.parser')
#print(soup.prettify()) #Observemos los cambios
```

Ahora que tenemos un código html ya limpio, podemos buscar dentro de éste, ya sea por atributo, etiqueta o clase.
Veamos un ejemplo:

In [24]:

```
comercial = soup.find_all('p') #Esto hará que la variable comercial almacene todos los atributos <
p>...</p>

#for x in comercial:
#    print(x) #Imprimimos nuestro botín.
```

Supongamos, por ejemplo, que queremos extraer el enunciado de la primera noticia de la página de ingeniería comercial uc. En ese caso lo podríamos hacer, mediante la siguiente acción.

In [25]:

```
primera = soup.find_all('p')[0].get_text()
print(primera)
```

```
El próximo martes 4 de septiembre la Facultad de Ciencias Económicas y Administrativas recibirá un
concierto en el que participará el estudiante de Ingeniería Comercial y pianista, Martín Cruzat ju
nto al reconocido violinista alemán, Manuel Druminski. Manuel Druminski ha recorrido el mundo de
la mano de su violín. Así ha tocado en diversas ciudades de […]
```

La misma acción de busqueda la podemos realizar tanto mediante id, como mediante clases. En estos casos simplemente alteramos un poco el método find.

In [35]:

```
por_identificador = list(soup.find(id="listado-contenido").children)
#print(por_identificador)

por_clase = soup.find_all(class_="grid_5 last")[0].get_text()
#print(por_clase)
```

Durante las próximas semanas estaremos trabajando estos métodos, interactuando también con expresiones regulares para poder realizar acciones más complejas. Por mientras, les recomiendo leer la API de la librería para aprender más al respecto.

Ejercicios

Ejercicio 1:

Escriba una función que invierta sus palabras. Es decir, que al ingresar una palabra como, por ejemplo, "espejo", esta devuelva la misma palabra, pero escrita al revés.

In [16]:

```
Escriba su palabra: join the navy
yvan eht nioj
```

Ejercicio 2:

Genere una función con cuatro parámetros que busque los números entre dos valores (incluyendolos) que sean divisibles por un número, pero no por otro (ambos dados). Cree la función de tal forma que si ningún parámetro es introducido, ésta busque los números entre 120 y 556 que son divisibles por 5, pero no por 10.

In [29]:

```
125,135,145,155,165,175,185,195,205,215,225,235,245,255,265,275,285,295,305,315,325,335,345,355,365
,385,395,405,415,425,435,445,455,465,475,485,495,505,515,525,535,545,555
5,15,25,35,45,55,65,75,85,95
```

Ejercicio 3

Descargue todos los links de la primera página de problemas de Kattis y almacenelos en un archivo .txt

Ejercicio 4

Importe la primera tabla del siguiente sitio web y, utilizando pandas, recreela hasta los primeros 10 países.

In [62]:

Out[62]:

		PIB
Posición	**País**	
	Mundo	79 865 481
1	**Estados Unidos**	20 412 870
	Unión Europea	19 669 743
2	**China**	14 092 514
3	**Japón**	5 167 051
4	**Alemania**	4 211 635
5	**Reino Unido**	2 936 286
6	**Francia**	2 925 096
7	**India**	2 848 231
8	**Italia**	2 181 970
9	**Brasil**	2 138 918
10	**Canadá**	1 798 512

Ejercicios Propuestos

Ejercicio 1

En base al ejercicio 3, cree una función que descargue todos los links de los problemas de Kattis. Es importante que sean solo links de los problemas y que se llegue hasta la última página, identificando correctamente cuál es esta.

In [62]:

```
```

```
La última página con ejercicios es la número 20
```

¿Dudas?