

Prácticas con fechas y horas

Servicios Telemáticos 2015-2016

Grado en Ingeniería Telemática

Universidad Rey Juan Carlos

<http://gsync.urjc.es>

Observaciones

- Crea el directorio `~/st/practica03` donde escribirás el resto de ficheros de esta práctica. Recuerda que la recogida de las prácticas se hará automáticamente el día del examen, así que es muy importante que respetes al pie de la letra todos los nombres de ficheros y directorios que indiquen todos los enunciados.

Práctica 3.1 Zonas horarias

Escribe un fichero `~/st/practica03/fechas.py` que

1. Aceptará desde la entrada estándar un fichero de texto que contendrá, en cada línea: el número de línea (empezando en 1) , un espacio y una fecha
2. El formato de la fecha podrá ser o bien *unix timestamp* o bien *aaaa-mm-dd hh:mm:ss, ciudad*
Donde ciudad podrá ser *Madrid, Londres, Moscu, Tokio, New_York* o *UTC* y cada línea podrá emplear un formato o ciudad distinto

Ejemplo:

```
1 1262684480
2 2010-10-02 13:28:10, Londres
3 2002-05-21 15:16:01, New_York
4 1436762748
5 2006-10-31 20:07:42, New_York
6 2005-12-22 21:32:40, Moscu
7 2013-07-08 02:42:16, Madrid
8 2010-02-19 10:58:03, UTC
9 2011-11-09 02:49:09, Londres
10 2005-01-24 20:09:56, Tokio
11 2011-10-05 07:27:22, Moscu
```

3. Si alguna línea no cumple con esta especificación, el script deberá mostrar un mensaje de error en `stderr` y morir
4. Si el script no recibe ninguna opción, o si recibe la opción `-t utc` o `--timezone utc` mostrará por salida estándar el número de línea (empezando por 1) y la hora UTC correspondiente a cada fecha de la entrada, con el formato del siguiente ejemplo:

```
1 2010-01-05 09:41:20+00:00
2 2010-10-02 12:28:10+00:00
3 2002-05-21 19:16:01+00:00
4 2015-07-13 04:45:48+00:00
5 2006-11-01 01:07:42+00:00
6 2005-12-22 18:32:40+00:00
7 2013-07-08 00:42:16+00:00
8 2010-02-19 10:58:03+00:00
9 2011-11-09 02:49:09+00:00
10 2005-01-24 11:09:56+00:00
```

Observa que este es el formato por omisión que emplea python al convertir un `datetime` con zona en cadena, no tendrás que personalizar el formato

5. Si el script recibe la opción `-t epoch` o `--timezone epoch` mostrará por salida estándar el número de línea (empezando por 1) y el timestamp unix correspondiente a cada fecha de la entrada

Ejemplo

```
1 1262684480
2 1286022490
3 1022008561
4 1436762748
5 1162343262
6 1135276360
7 1373244136
8 1266577083
9 1320806949
10 1106564996
```

6. Si el script recibe la opción `-t madrid` o `--timezone madrid` mostrará por salida estándar el número de línea (empezando por 1) y la hora local de madrid, con el formato del siguiente ejemplo:

```
1 2010-01-05 10:41:20+01:00
2 2010-10-02 14:28:10+02:00
3 2002-05-21 21:16:01+02:00
4 2015-07-13 06:45:48+02:00
5 2006-11-01 02:07:42+01:00
6 2005-12-22 19:32:40+01:00
7 2013-07-08 02:42:16+02:00
8 2010-02-19 11:58:03+01:00
9 2011-11-09 03:49:09+01:00
10 2005-01-24 12:09:56+01:00
11 2011-10-05 05:27:22+02:00
```

7. De la misma manera, el script podrá recibir con las opción `-t --timezone` las cadenas *londres*, *moscu*, *tokio*, *new_york* (observa que son minúsculas) y el comportamiento será análogo al anterior.
8. No es necesario que te preocupes por el caso en que el usuario introduzca más de una zona horaria. Por ejemplo `-t londres -t tokio`, o `-t londres tokio`. Es aceptable que en este caso la librería que procesa los argumentos genere una excepción. O que simplemente se acepte la última zona.

Usa el fichero `fechas.txt`¹ para probar tu script, y compara la salida con los resultados que encontrarás en `solucionfechas.txt`² para comprobar que se comporta correctamente

¹<http://gsyc.es/~mortuno/st/fechas.txt>

²<http://gsyc.es/~mortuno/st/solucionfechas.txt>