Paradigmas de Programación

Sesión 3 Laboratorio

Cristian Tejedor García
Departamento de Informática
Universidad de Valladolid

Curso 2016-17

Grado en Ingeniería Informática
INDAT

Contenido

1. Objetivos
2. Cadenas de caracteres en Python
3. Reto (ayuda)

1. Objetivos

- Resolver dudas teoría y ejercicios de la sesión 2.
- Explicar las funciones de cadenas de caracteres en Python.
- Realizar ejercicios de repaso.

2. Cadenas de caracteres (I)


```
# 1. Longitud
print len("cadena") # True

# 2. Comparación (tener cuidado con mayúsculas y minúsculas)
print "cadena" == "cadena"  # True
print "cadena" != "Cadena"  # True
print "cadena".lower() == "Cadena".lower()  # True

# 3. División de elementos
a = "AMPLI 451 PARAD 729"
b = a.split(' ') # Delimitador, puede ser cualquier elemento
print b  # ['AMPLI', '451', 'PARAD', '729']
```

2. Cadenas de caracteres (II)


```
# 4. Búsqueda
# find() devuelve -1 si no encuentra, index() una excepción
print "cadena".find('e')  # 3
print "cadena".find('una')  # -1
print 'i' in "cadena"  # False

# 5. Acceso
print "cadena"[0]  # c
print "cadena"[-1]  # a
print "cadena"[-2]  # n

# 6. Concatenación
print "cadena" + "s largas"  # cadenas largas
```

2. Cadenas de caracteres (III)


```
# 7. Slicing (subcadenas)
print "cadena"[0:]  # cadena
print "cadena"[:-2]  # cade
print "cadena"[2:-2]  # de
# Empieza:Acaba-1:número_saltos
print "cadena muy larga"[2:-2:3]  # daul

# 8. Reemplazamiento
# Devuelve una copia, porque String es inmutable
print "cadena larga".replace("a", "W", 3)  # cWdenW lWrga

# 9. Conteo
print "cabeza".count('a') # 2
```

2. Cadenas de caracteres (IV)

 # 10. Última ocurrencia de un caracter (fórmula) *lastIndexOf() en Java*

```
cadena = "Paradigmas es un grupo ejemplar"
print len(cadena) # 31
print len(cadena) - cadena[::-1].find('e') – 1 # 25, cadena[::-1] invierte cadena
```

 # 11. Eliminación de todos los espacios al inicio y fin

```
print '        Hello        '.strip()  # Hello
```

 # 12. Comienzo/fin por

```
print "Casa del campo".startswith("Casa")  # True
print "Casa del campo".endswith("ampa")  # False
```

2. Cadenas de caracteres (V)


```
# 13. Expresiones regulares (subcadenas con cierta forma)
# https://platzi.com/blog/expresiones-regulares-python/

import re

# Coincide con una letra, seguida de al menos 1 dígito entre 2 y 6
patron = re.compile('a[2-6]+')
cadena = 'ba3425 a6'  # La coincidencia del patrón no está al principio!

# search() busca en la cadena alguna ocurrencia del patrón
print patron.search(cadena)  # <_sre.SRE_Match object at 0x0311C3D8>

# match() igual que search() pero coincidencia sólo al comienzo de la cadena
print patron.match(cadena)  # None

# findall() devuelve todas las ocurrencias en forma de lista
print patron.findall(cadena)  # ['a3425', 'a6']
```

3. Reto (ayuda)

Disponemos de un **fichero** de texto que contiene la información de la matrícula de alumnos de primer curso. **Cada línea** del fichero representa un par **asignatura-alumno** (la asignatura se representa por una abreviatura de 2, 3 o 4 letras, le sigue un espacio y el NIA del alumno): https://goo.gl/LgfLbG

Crear una **función reto**, que lea el fichero y lo procese de forma que devuelva un diccionario cuyo índice sea una tupla de dos asignaturas y que almacene el número de alumnos que están matriculados simultáneamente de ambas asignaturas.

Aproximadamente: 25 líneas en Python ☺ 100 líneas en Java ☹

Ejemplo de uso: dic = reto("reto.txt“, ("PAR","AMAT"))

```
dic[("PAR","AMAT")] - 113  # Devuelve el número de alumnos matriculados
# simultáneamente en Paradigmas y Ampliación de Matemáticas

# Paso 1: Lista de pares ("ASIG",'NIA') del fichero
# Paso 2: Diccionario con entradas: "ASIG":['NIA','NIA']
# Paso 3: Lista de asignaturas (sin repetir)
# Paso 4: Diccionario con entradas: ("ASIG1","ASIG2"):['NIA', 'NIA'…'NIA']
```

Para afianzar…

- Ejercicios de la asignatura (sesión 2) ~2horas:
  - https://www.infor.uva.es/~cvaca/asigs/docpar/sesion2.pdf
- Ejercicios de la asignatura (sesión 6) ~2.5horas:
  - https://www.infor.uva.es/~cvaca/asigs/docpar/sesion6.pdf
- Reto propuesto sesión 2.
- “Python para todos”, Raúl González Duque
  - http://mundogeek.net/tutorial-python/
  - **Capítulos**: Revisitando objetos y repaso de todos los anteriores.