Tipos de datos estructurados

Ricardo Pérez López

IES Doñana, curso 2019/2020

Índice general

1. Secuencias

1.1. Concepto de secuencia

- Una secuencia es una estructura de datos que:
  - permite el acceso eficiente a sus elementos usando índices enteros, y
  - se le puede calcular su longitud mediante la función `len`.
- Las secuencias se dividen en:

  - **Inmutables**: cadenas (`str`), tuplas (`tuple`), rangos (`range`), conjuntos congelados (`frozenset`).
  - **Mutables**: listas (`list`), diccionarios (`dict`), conjuntos (`set`).

1.2. Operaciones comunes

1.2.1. En cualquier tipo de secuencia

- Todas las secuencias (ya sean cadenas, listas, tuplas o rangos) comparten un conjunto de operaciones comunes.
- Además de estas operaciones, las secuencias del mismo tipo admiten comparaciones. Las tuplas y las listas se comparan lexicográficamente elemento a elemento.
  - Eso significa que dos secuencias son iguales si cada elemento es igual y las dos secuencias son del mismo tipo y tienen la misma longitud.
- La siguiente tabla enumera las operaciones sobre secuencias, ordenadas por prioridad ascendente. `s` y `t` son secuencias del mismo tipo, `n`, `i`, `j` y `k` son enteros y `x` es un dato cualquiera que cumple con las restricciones que impone `s`.

Operación	Resultado
x `in` *s*	`True` si algún elemento de *s* es igual a *x*
x `not in` *s*	`False` si algún elemento de *s* es igual a *x*
s `+` *t*	La concatenación de *s* y *t*
s `*` *n*	Equivale a añadir *s* a sí mismo *n* veces
n `*` *s*	
s`[`*i*`]`	El *i*-ésimo elemento de *s*, empezando por 0
s`[`*i*`:`*j*`]`	Rodaja de *s* desde *i* hasta *j*
s`[`*i*`:`*j*`:`*k*`]`	Rodaja de *s* desde *i* hasta *j* con paso *k*
`len(`*s*`)`	Longitud de *s*
`min(`*s*`)`	El elemento más pequeño de *s*
`max(`*s*`)`	El elemento más grande de *s*
s`.index(`*x*`[, `*i*`[,`*j*`]])`	El índice de la primera aparición de *x* en *s* (desde el índice *i* inclusive y antes del *j*)
s`.count(`*x*`)`	Número total de apariciones de *x* en *s*

1.2.2. En secuencias mutables

- En la siguiente tabla, `s` es una instancia de un tipo de secuencia mutable, `t` es cualquier dato iterable y `x` es un dato cualquiera que cumple con las restricciones que impone `s`:

Operación	Resultado
s`[`*i*`] =` *x*	El elemento *i* de *s* se sustituye por *x*
s`[`*i*`:`*j*`] =` *t*	La rodaja de *s* desde *i* hasta *j* se sustituye por el contenido del iterable *t*

Operación	Resultado
`del` *s*`[`*i*`:`*j*`]`	Igual que *s*`[`*i*`:`*j*`] = []`
s`[`*i*`:`*j*`:`*k*`]` = *t*	Los elementos de *s*`[`*i*`:`*j*`:`*k*`]` se sustituyen por los de *t*
`del` *s*`[`*i*`:`*j*`:`*k*`]`	Elimina de la secuencia los elementos de *s*`[`*i*`:`*j*`:`*k*`]`

Operación	Resultado
s`.append(`*x*`)`	Añade *x* al final de la secuencia; es igual que *s*`[len(`*s*`):len(`*s*`)] = [`*x*`]`
s`.clear()`	Elimina todos los elementos de *s*; es igual que `del` *s*`[:]`
s`.copy()`	Crea una copia *superficial* de *s*; es igual que *s*`[:]`
s`.extend(`*t*`)`	Extiende *s* con el contenido de *t*; es como hacer *s*`[len(`*s*`):len(`*s*`)] = `*t*
s `+=` *t*	
s `*=` *n*	Modifica *s* repitiendo su contenido *n* veces
`max(`*s*`)`	El elemento más grande de *s*
s`.insert(`*i*`,`*x*`)`	Inserta *x* en *s* en el índice *i*; es igual que *s*`[`*i*`:`*i*`] = [`*x*`]`
s`.pop([`*i*`])`	Extrae el elemento *i* de *s* y lo devuelve (por defecto, *i* vale −1)
s`.remove(`*x*`)`	Quita el primer elemento de *s* que sea igual a *x*
s`.reverse()`	Invierte los elementos de *s*

1.3. Cadenas (`str`)

1.3.1. Operadores

1.3.1.1. Concatenación

1.3.1.2. Repetición

1.3.1.3. Indexación

1.3.1.4. *Slicing*

1.3.2. Funciones

1.3.3. Métodos

1.3.4. Expresiones regulares

1.4. Listas