

Clase 3

6. Conjuntos y Diccionarios

Existen 2 categorías de variables para agrupación de datos: Mappings y Containers.

La única implementación de mappings en python es el diccionario mientras que de containers se cuenta con:

- List
- Set
- FrozenSet
- Tuple
- *Collections*

6.1 Tuplas

Las tuplas se pueden construir de diferentes maneras:

- Usando un par de símbolos de paréntesis, para indicar una tupla vacía: ()
- Usando una coma al final, para crear una tupla de un único elemento: a, o (a,)
- Separando los elementos por comas: a, b, c o (a, b, c)
- Usando la función incorporada tuple(): tuple() o tuple(iterable)

El constructor genera una tupla cuyos elementos son los mismos y están en el mismo orden que los elementos del iterable, donde iterable puede ser una secuencia, un contenedor que soporta iteración, o un objeto de tipo iterator. Si iterable es ya de por sí una tupla, se retorna sin cambiar. Por ejemplo, tuple('abc') retorna ('a', 'b', 'c') y tuple([1, 2, 3]) retorna (1, 2, 3). Si no se indica ningún parámetro, el constructor creará una nueva tupla vacía. ().

Nótese que es la coma la que realmente construye la tupla, no los paréntesis. Los paréntesis son opcionales, excepto en el caso de la tupla vacía, o cuando se necesitan para evitar una ambigüedad sintáctica. Por ejemplo, f(a, b, c) es una llamada a una función con tres parámetros, pero f((a, b, c)) es una llamada a una función con un único parámetro, en este caso una tupla de tres elementos.

6.2 Conjuntos

Un objeto de tipo set es una colección no ordenada de distintos objetos hashable. Los casos de uso habituales incluyen comprobar la pertenencia al conjunto de un elemento, eliminar duplicados de una secuencia y realizar operaciones matemáticas como la intersección, la unión, la diferencia o la diferencia simétrica. (Para otros tipos de contenedores véanse las clases incorporadas dict, list, y tuple, así como el módulo collections).

Como otras colecciones, los conjuntos soportan `x in set`, `len(set)` y `for x in set`. Como es una

colección sin orden, los conjuntos no registran ni la posición ni el orden de inserción de los elementos. Por lo mismo, los conjuntos no soportan indexado, ni operaciones de segmentado, ni otras capacidades propias de las secuencias.

En la actualidad hay dos tipos de conjuntos incorporados: set y frozenset. La clase set es mutable, es decir, el contenido del conjunto puede ser modificado con métodos como add() y remove(). Como es mutable, no tiene un valor de hash y no pueden ser usados como claves de diccionarios ni como elementos de otros conjuntos. La clase frozenset es inmutable y hashable, es decir, que sus contenidos no pueden ser modificados después de creados. Puede ser usado, por tanto, como claves de diccionario o como elemento de otro conjunto.

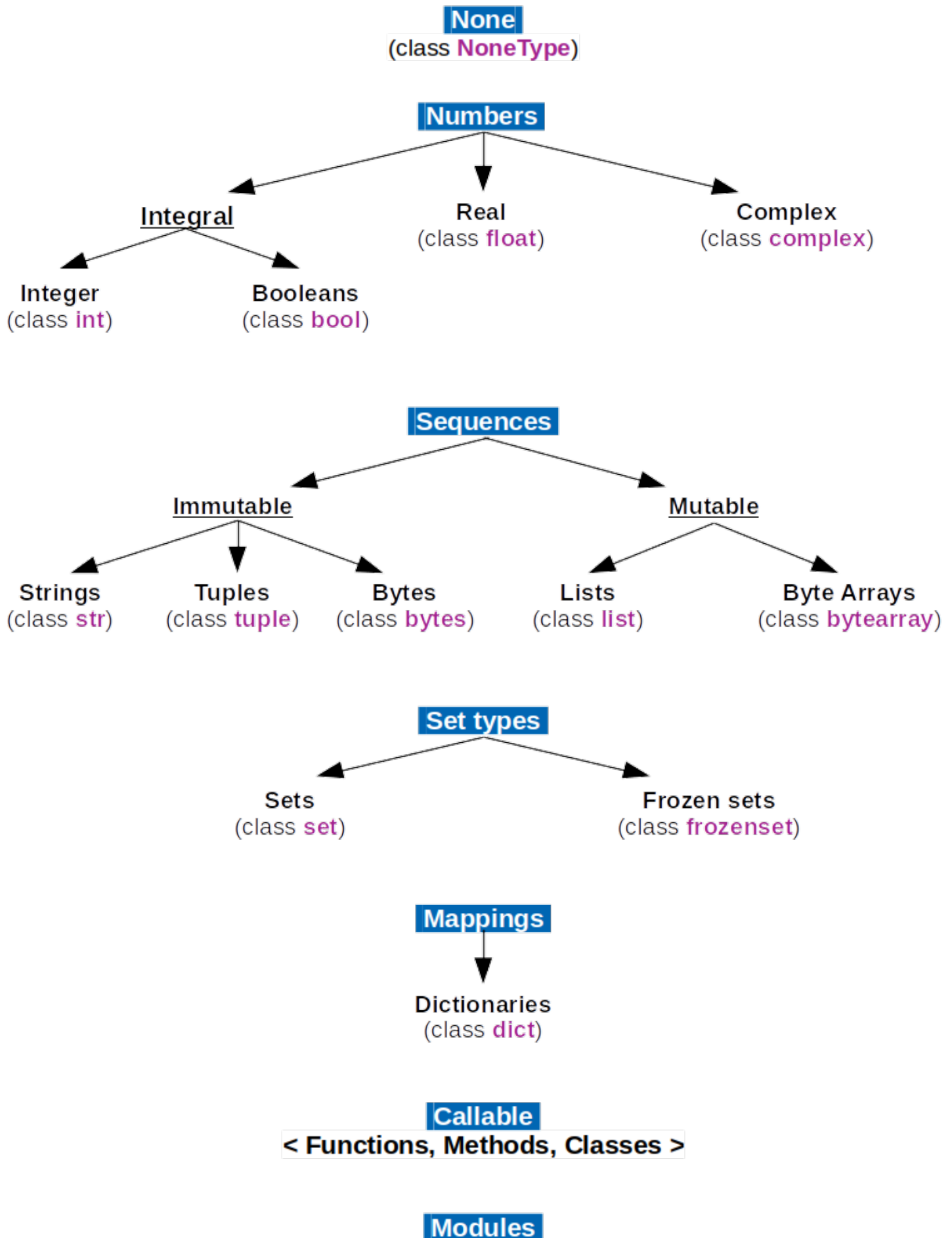
Se pueden crear conjuntos no vacíos (sets, no frozensets) escribiendo una lista de elementos separados por comas, entre llaves, por ejemplo {'jack', 'sjoerd'}, además de con el constructor de la clase set.

6.3 Diccionarios

Un objeto de tipo mapping relaciona valores (que deben ser hashable) con objetos de cualquier tipo. Los mapas son objetos mutables. En este momento solo hay un tipo estándar de mapa, los dictionary. (Para otros tipos contenedores, véanse las clases incorporadas list, set, y tuple, así como el módulo collections).

Python 3

The standard type hierarchy



9. Estructuras de datos

Formato de organización, administración y almacenamiento de datos.

9.1 Listas

Colección lineal de nodos, donde cada nodo contiene un valor y una referencia al siguiente nodo.

9.2 Pilas

9.3 Colas

9.4 Árboles

Ejercicios

Implemente una de las siguientes tres estructuras de datos en python (Pilas, colas árboles) con las siguientes funciones básicas:

- agregar
- eliminar
- buscar

Definiciones

hashable

Un objeto es hashable si tiene un valor de hash que nunca cambiará durante su tiempo de vida (necesita un método **hash()**), y puede ser comparado con otro objeto (necesita el método **eq()**). Los objetos hashables que se comparan iguales deben tener el mismo número hash.

Ser hashable hace a un objeto utilizable como clave de un diccionario y miembro de un set, porque éstas estructuras de datos usan los valores de hash internamente.

La mayoría de los objetos inmutables incorporados en Python son hashables; los contenedores mutables (como las listas o los diccionarios) no lo son; los contenedores inmutables (como tuplas y conjuntos frozensets) son hashables si sus elementos son hashables . Los objetos que son instancias de clases definidas por el usuario son hashables por defecto. Todos se comparan como desiguales (excepto consigo mismos), y su valor de hash está derivado de su función **id()**.

Fuentes

La sección 6 fué extraida de la documentación oficial de python. Vease docs.python.org.