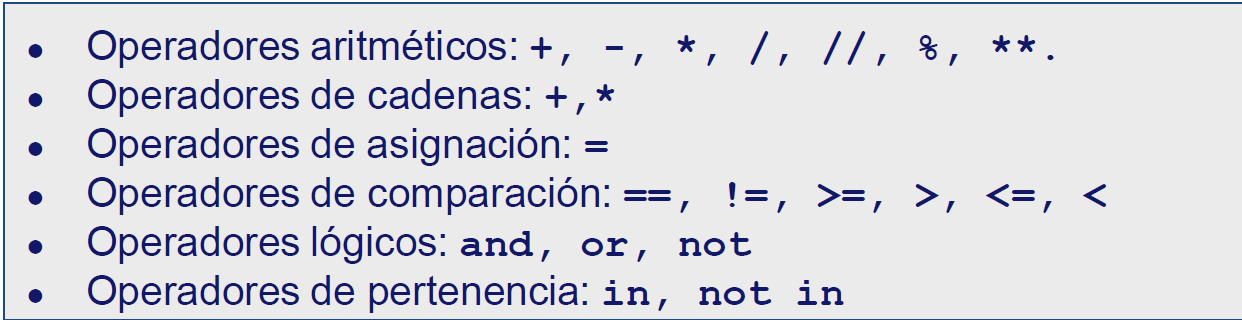
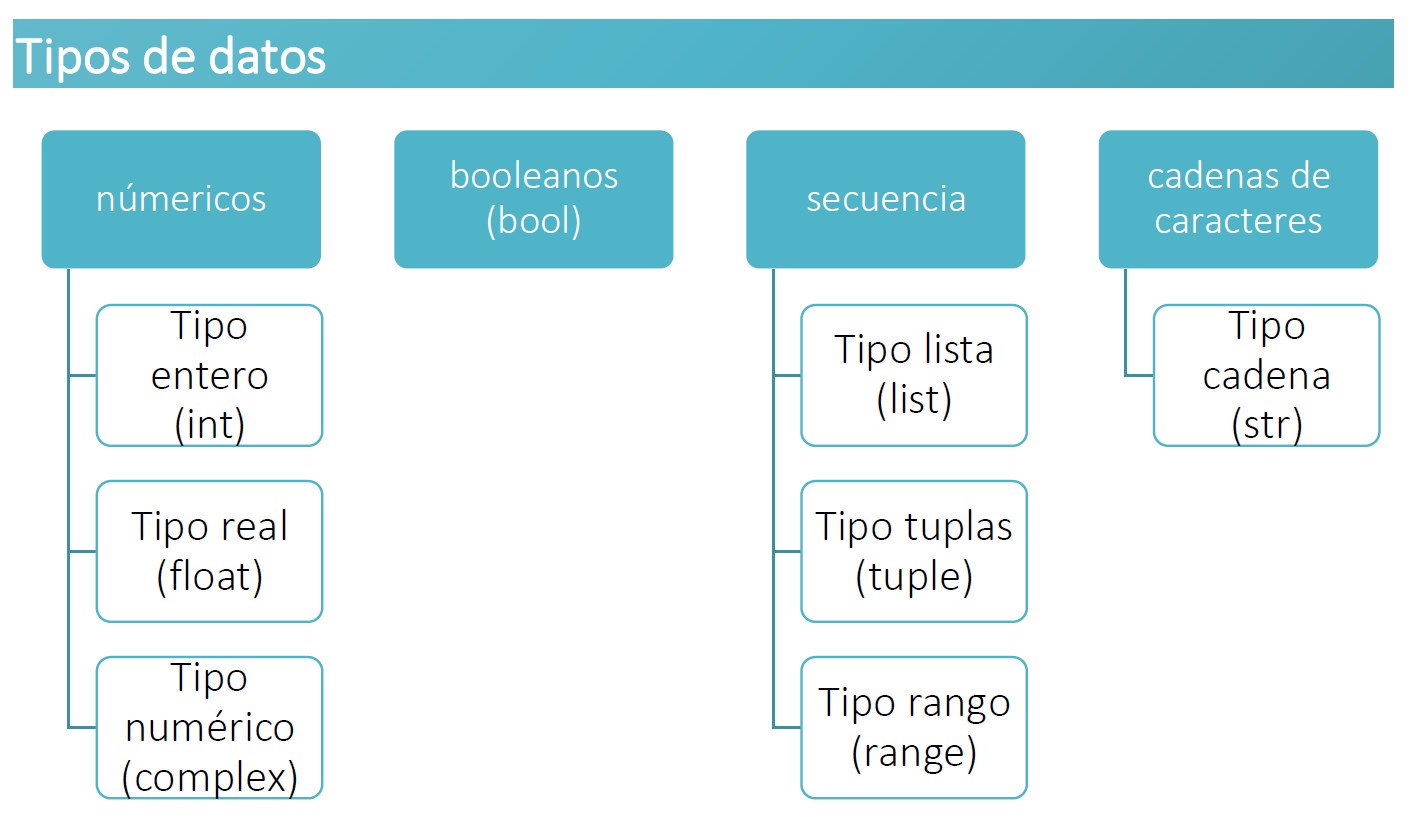
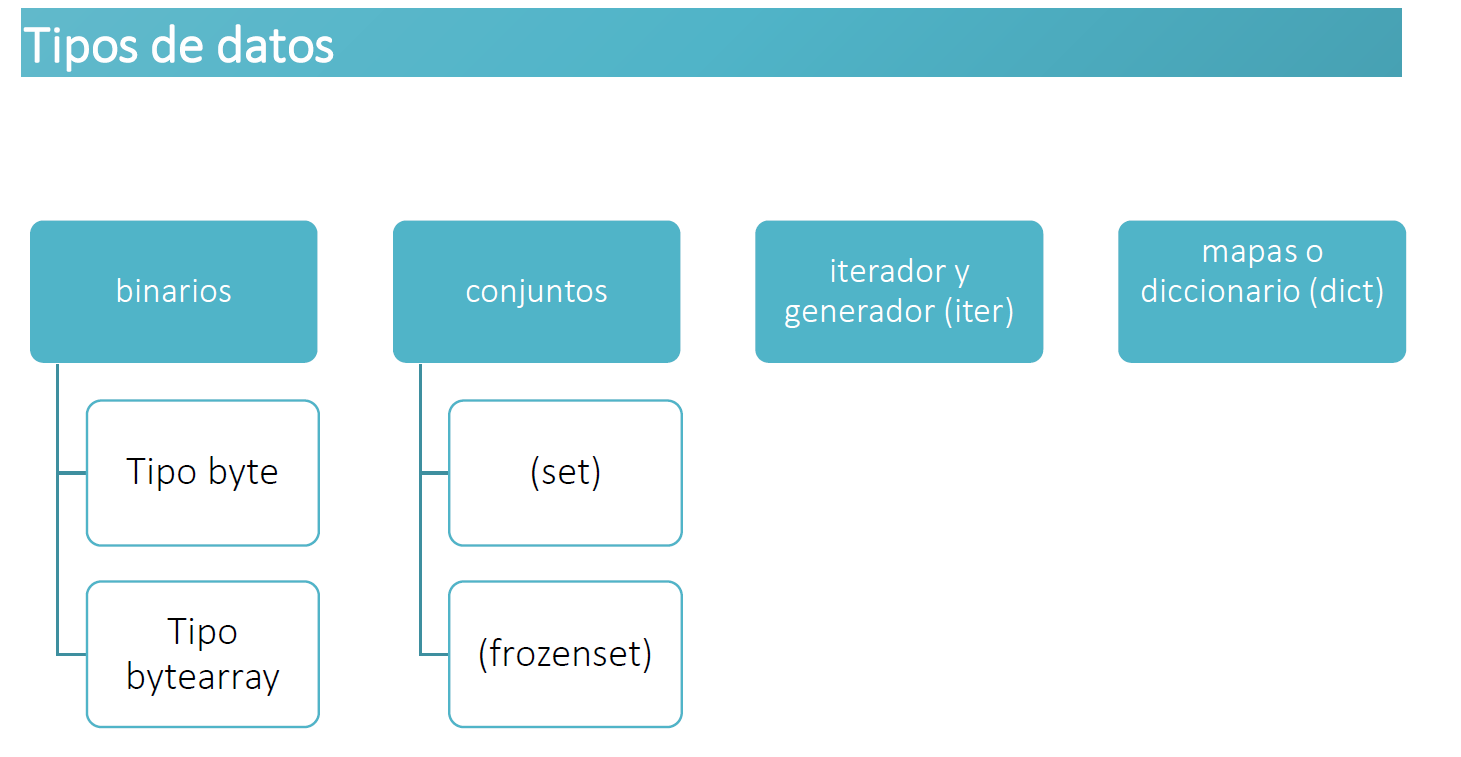
**Operadores Integradas en Python**

****

1. Los paréntesis rompen la precedencia.
2. La potencia (\*\*)
3. Operadoresunarios(+ -)
4. Multiplicar, dividir, módulo y división entera (\* % // )
5. Suma y resta (+ -)
6. Operador binario AND (&)
7. Operadores binario OR y XOR (^ |)
8. Operadores de comparación (<= < > >=)
9. Operadores de igualdad(== !=)
10. Operadores de asignación (=)
11. Operadores de pertenencia (in, in not)
12. Operadores lógicos (not, or, and)

****

****

**Funciones Integradas en Python**

**🔢 Funciones Matemáticas**

* **abs(x)** → Devuelve el valor absoluto de x.
* **divmod(a, b)** → Devuelve un par (a // b, a % b), cociente y residuo de la división.
* **float(x)** → Convierte x a número decimal (coma flotante).
* **int(x, base=10)** → Convierte x a un número entero, con base opcional.
* **max(iterable, \*args, key=None)** → Devuelve el valor más grande.
* **min(iterable, \*args, key=None)** → Devuelve el valor más pequeño.
* **pow(x, y, mod=None)** → Calcula x\*\*y, opcionalmente con un módulo.
* **round(number, ndigits=None)** → Redondea un número.
* **sum(iterable, start=0)** → Suma los elementos de un iterable.

**🄦 Funciones de Conversión de Tipo**

* **bool(x)** → Convierte x en True o False.
* **bytes(x, encoding, errors)** → Convierte x en bytes.
* **bytearray(x, encoding, errors)** → Similar a bytes, pero mutable.
* **chr(i)** → Convierte un código Unicode en su carácter.
* **ord(c)** → Convierte un carácter en su código Unicode.
* **complex(real, imag)** → Crea un número complejo.
* **dict()** → Crea un diccionario.
* **list(iterable)** → Convierte un iterable en una lista.
* **tuple(iterable)** → Convierte un iterable en una tupla.
* **set(iterable)** → Convierte un iterable en un conjunto.
* **frozenset(iterable)** → Como set, pero inmutable.
* **str(x)** → Convierte x a cadena de texto.
* **ascii(x)** → Similar a repr(), pero con caracteres ASCII legibles.
* **repr(x)** → Representación en cadena de x.
* **hex(x)** → Convierte un número en hexadecimal.
* **oct(x)** → Convierte un número en octal.
* **bin(x)** → Convierte un número en binario.
* **memoryview(x)** → Crea una vista en memoria de un objeto.

**📊 Funciones para Estructuras de Datos**

* **enumerate(iterable, start=0)** → Devuelve índices y valores del iterable.
* **filter(function, iterable)** → Filtra elementos que cumplen con function.
* **map(function, iterable)** → Aplica function a cada elemento.
* **zip(\*iterables)** → Une elementos de iterables en tuplas.

**🛠️ Funciones para Manipulación de Atributos y Métodos**

* **getattr(obj, name, default)** → Obtiene un atributo de un objeto.
* **setattr(obj, name, value)** → Asigna un valor a un atributo.
* **hasattr(obj, name)** → Verifica si un objeto tiene un atributo.
* **delattr(obj, name)** → Elimina un atributo de un objeto.
* **property(fget, fset, fdel, doc)** → Crea una propiedad en una clase.
* **classmethod(func)** → Convierte una función en un método de clase.
* **staticmethod(func)** → Convierte una función en un método estático.
* **super()** → Llama métodos de una clase padre.

**🔍 Funciones de Inspección**

* **dir(obj)** → Lista los atributos de un objeto.
* **globals()** → Devuelve el diccionario de variables globales.
* **locals()** → Devuelve el diccionario de variables locales.
* **vars(obj)** → Devuelve los atributos de un objeto.
* **id(obj)** → Devuelve el identificador único del objeto.
* **type(obj)** → Devuelve el tipo del objeto.
* **isinstance(obj, classinfo)** → Verifica si obj es instancia de classinfo.
* **issubclass(cls, classinfo)** → Verifica si cls es subclase de classinfo.

**♻️ Iteradores y Secuencias**

* **iter(obj, sentinel=None)** → Crea un iterador.
* **next(iterator, default=None)** → Devuelve el siguiente elemento del iterador.
* **len(obj)** → Retorna la cantidad de elementos.
* **sorted(iterable, key=None, reverse=False)** → Ordena elementos.
* **reversed(seq)** → Devuelve un iterador en orden inverso.
* **slice(start, stop, step)** → Crea un objeto de segmentación.
* **range(start, stop, step)** → Genera una secuencia de números.

**🖨️ Entrada y Salida**

* **print(\*objects, sep=' ', end='\n', file=sys.stdout)** → Imprime en consola.
* **input(prompt)** → Recibe una entrada del usuario.
* **open(file, mode, encoding)** → Abre un archivo.

**🛡️ Funciones de Evaluación y Ejecución**

* **eval(expression, globals=None, locals=None)** → Evalúa una expresión Python.
* **exec(object, globals=None, locals=None)** → Ejecuta código Python.
* **compile(source, filename, mode)** → Compila código fuente en un objeto ejecutable.
* **\_\_import\_\_(name, globals=None, locals=None, fromlist=(), level=0)** → Importa un módulo (similar a import).