

Unidad 2

Tipos y estructuras de datos



En Python existen varios tipos de datos compuestos estándar disponibles por defecto en el intérprete, como los tipos numéricos, secuencias, mapeos y conjuntos usados para agrupar otros valores.

Para el caso de las estructuras de datos se usan variables y constantes las cuales usan operadores para tratar los tipos de datos estándar.

En esta lección se describen las variables, operadores y sus tipos de datos en el lenguaje Python.

1. Jerarquía de tipos estándar.

A continuación, se muestra una lista de los tipos que están integrados en Python.

En Python tiene varios tipos de datos compuestos estándar disponibles por defecto en el intérprete, como los tipos numéricos, secuencias, mapeos y conjuntos usados para agrupar otros valores, para el caso de las estructuras de datos se usan variables y constantes las cuales usan operadores para tratar los tipos de datos estándar.

Clasificación

Los tipos de datos compuestos estándar se pueden clasificar como los dos siguientes:

- **Mutable:** su contenido (o dicho valor) puede cambiarse en tiempo de ejecución.
- **Immutable:** su contenido (o dicho valor) no puede cambiarse en tiempo de ejecución.

Se pueden resumir los tipos de datos compuestos estándar en la siguiente tabla:

Categoría de tipo	Nombre	Descripción
Números inmutables	int	entero
	long	entero long
	float	coma flotante
	complex	complejo
	bool	booleano
Secuencias inmutables	str	cadena de caracteres
	unicode	cadena de caracteres Unicode
	tuple	tupla
	xrange	rango inmutable
Secuencias mutables	list	lista
	range	rango mutable
Mapeos	dict	diccionario
Conjuntos mutables	set	conjunto mutable
Conjuntos inmutables	frozenset	conjunto inmutable

2. Variables, constantes y palabras reservadas.

2.1. Variables.

Es un nombre que se refiere a un objeto que reside en la memoria. El objeto puede ser de alguno de los tipos vistos (número o cadena de texto), o alguno de los otros tipos existentes en Python.

Cada variable debe tener un nombre único llamado identificador. Es de mucha ayuda pensar las variables como contenedores que en su interior llevan datos los cuales pueden ser cambiados a través de técnicas de programación.

Alcance de las variables

Las variables en Python son locales por defecto. Esto quiere decir que las variables definidas y utilizadas en el bloque de código de una función, sólo tienen existencia dentro de la misma, y no interfieren con otras variables del resto del código.

Ejemplos de variables.

Área de codificación.

```
1  edad = 11 #declaramos variable de tipo int=entero
2  NOMBRE = "pascual" #declaramos constante de tipo str=cadena
3
4  print(type(edad)) #imprimimos tipo de variable edad
5  print(type(NOMBRE)) #imprimimos tipo de constante nombre
6
7  print(NOMBRE, "tu edad es: ", edad) #imprimimos edad pascual
8
9  edad = 21 #cambiamos edad de pascual
10
11 print(NOMBRE, "tu edad es: ", edad) #imprimimos nueva edad
```

Área de ejecución.

```
<class 'int'>
<class 'str'>
pascual tu edad es:  11
pascual tu edad es:  21
```

2.2. Constante.

Una constante es un tipo de dato el cual no puede ser cambiada. Es de mucha ayuda pensar que las constantes son como contenedores que en su interior llevan datos los cuales no pueden ser cambiados a través de técnicas de programación.

En Python, las constantes son usualmente declaradas y asignadas en letras mayúsculas, tal como se evidencia en el anterior ejemplo.

2.3. Palabras reservadas.

Las palabras reservadas de Python son las siguientes:

Reservas del Sistema	and, del, for, is, raise, assert, if, else, elif, from, lambda, return, break, global, not, try, class, except, or, while, continue, exec, import, yield, def, finally, in, print
-----------------------------	---

En total son 28 palabras reservadas, estas palabras no pueden utilizarse para nombrar variables.

Al igual que todos los lenguajes de programación, Python también cuenta con los siguientes operadores aritméticos.

Operador	Significado
+	Suma (o adición)
-	Resta (o sustracción)
*	Multiplicación
**	Exponente
%	Módulo (o resto)
/	División
//	División entera

3. Operadores de asignaciones.

Existe en Python todo un grupo de operadores los cuales le permiten básicamente asignar un valor a una variable, usando el operador (=).

Con estos operadores pueden aplicar la técnica denominada asignación aumentada.

Operador =

El operador igual a, (=), es el más simple de todos y asigna a la variable del lado izquierdo cualquier variable o resultado del lado derecho.

```
1 num = 2
2
3 print (num)
```

2
✖

Operador +=

El operador (+=) suma a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num += 1
3 print (num)
```

3
✖

Operador -=

El operador (-=) resta a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num -= 1
3 print (num)
```

1
✖

Operador *=

El operador (*=) multiplica a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num *= 2
3 print (num)
```

4
✖

Operador /=

El operador (/=) divide a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num /= 2
3 print (num)
```

1.0
✖

Operador **=

El operador (**=) calcula el exponente a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num **= 2
3 print (num)
```

4

**Operador //=**

El operador (//=) calcula la división entera a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num //= 2
3 print (num)
```

1

**Operador %=**

El operador (%=) devuelve el módulo de la división a la variable del lado izquierdo el valor del lado derecho.

```
1 num = 2
2 num %= 2
3 print (num)
```

0

