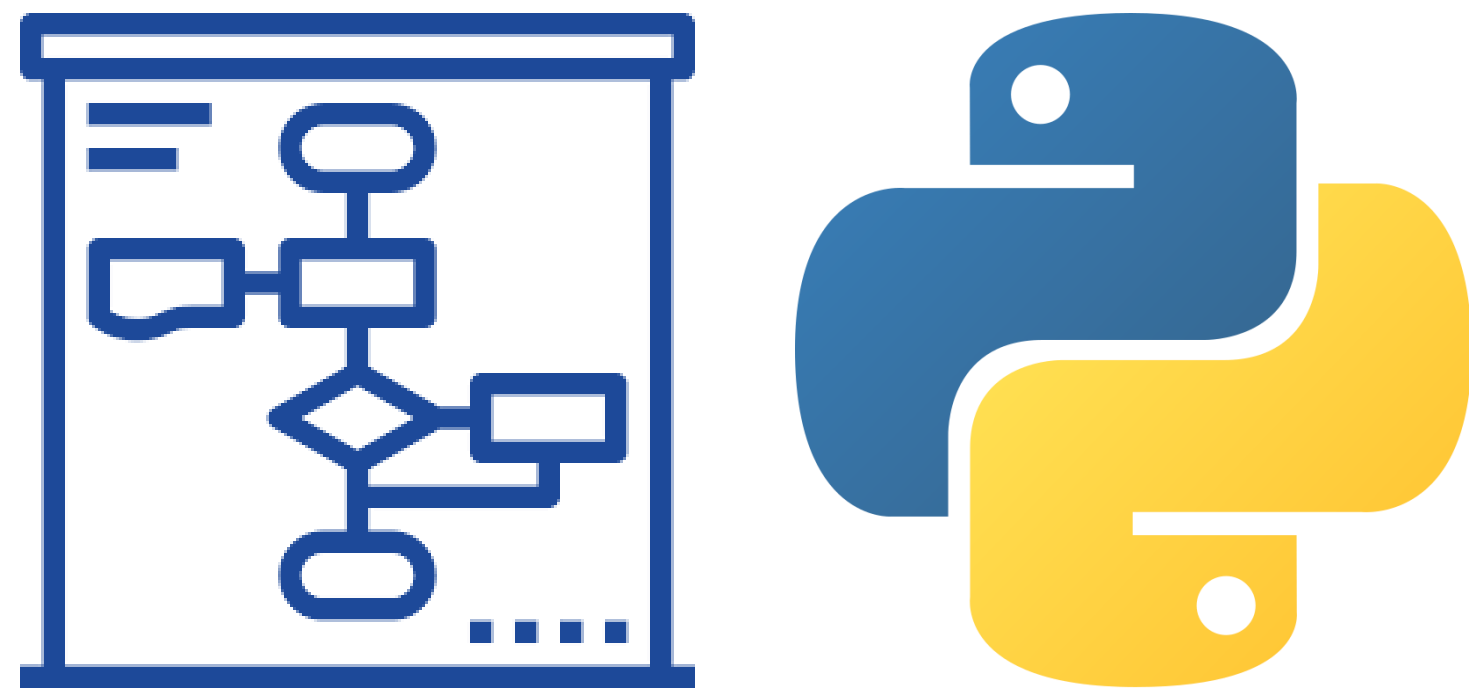




INTRODUCCIÓN A LA PROGRAMACIÓN



OTOÑO, 2022

UNIVERSIDAD TECNOLÓGICA DE CHILE
INSTITUTO PROFESIONAL
CENTRO DE FORMACIÓN TÉCNICA





UNIDAD II

Lenguajes Altamente Dinámicos/ Débilmente Tipados

Uso de módulos Python mediante import

¿Qué son los módulos de Python?

Python incluye una gran colección de módulos. Hay más de 200 módulos para tareas de programación comunes (Módulos numéricos y matemáticos, Módulos de programación funcional, Acceso a archivos y directorios, Servicios genéricos del sistema operativo, entre otros).

Estos módulos son llamados biblioteca estándar, puedes encontrarlos en la mayoría de las instalaciones en todas las plataformas, como un archivo que contiene un conjunto de funciones que desea incluir en su aplicación.

Estos módulos se pueden usar importándolos utilizando la sentencia `import`.

Nota: Podemos también crear nuestro archivo módulo/biblioteca, dentro del cual se pueden definir funciones, clases y variables, para enviarlos y compartirlos.



Nos permite importar toda la funcionalidad de un módulo de la Biblioteca estándar de Python

Sintaxis

- ▶ Utilizamos palabra clave import y seguidamente el nombre del módulo:

Con esa sintaxis importará el módulo matemático. Cualquier función que use debería usarse como `math.sqrt()`, `math.factorial()`, etc.

```
import math
```

- ▶ Para acceder solo a la función `randint` del módulo `random` en Python lo podemos expresar con la siguiente sintaxis:

```
from random import randint
```

- ▶ Para importar más de una funcionalidad.

```
from math import sqrt, pow
```

- ▶ Para definir un alias a una funcionalidad.

```
from math import sqrt as raiz, pow as elevar
```

- ▶ Para importar todas las funciones de matemáticas, Cualquier función que utilice se puede utilizar como `sqrt`, `factorial`, etc.

```
from math import *
```

Nota: Para ver qué funciones están disponibles:

```
import math  
print(dir(math))
```

Módulo math

Python tiene un conjunto de funciones matemáticas integradas, incluido un módulo matemático extenso, que permite realizar tareas matemáticas con números.

Sintaxis funciones matemáticas integradas

Las funciones min () y max () se pueden usar para encontrar el valor más bajo o más alto en un iterable:	<pre>x = min(5, 10, 25) y = max(5, 10, 25) print(x) print(y)</pre>
La función pow() devuelve el valor de un número a la potencia dada:	<pre>x = pow(4, 3) print(x)</pre>

```
a=max("Python")
print(a)
print(ord(a))
```



$$\sqrt[3]{100}$$

Nota: Con la función ord() obtenemos el valor ASCII de los caracteres.
Con la función chr() obtenemos el carácter ASCII asociado al valor.

Sintaxis funciones matemáticas con import

El método math.sqrt () devuelve la raíz cuadrada de un número:	<pre>import math x = math.sqrt(64) print()</pre>
La función trunc() devuelve la parte entera de diferentes números:	<pre>import math print(math.trunc(2.77)) print(math.trunc(8.32)) print(math.trunc(-99.29))</pre>

Escenario

Calcular el área y el perímetro de un triángulo conociendo el largo de los 3 lados.

las fórmulas a aplicar serían:

$$P = L1 + L2 + L3$$

$$S = P / 2$$

$$A = \sqrt{S \times (S - L1) \times (S - L2) \times (S - L3)}$$

Datos de entrada:

$$L1 = 4$$

$$L2 = 4$$

$$L3 = 5$$

Resultado:

$$\text{Perímetro} = 12$$

$$\text{Área} = 7.81$$

(redondear a 2 decimales)

Módulo random

Python tiene un módulo incorporado que puede usar para generar números aleatorios.

Sintaxis random

El método random() devuelve un número flotante aleatorio entre 0 y 1.	<pre>import random print(random.random())</pre>
El método randrange() devuelve un elemento seleccionado aleatoriamente del rango especificado, se puede especificar un paso.	<pre>import random print(random.randrange(3, 9)) #devuelve un número entre 3 (incluido) y 9 (no incluido)</pre>
El método randint() devuelve un elemento seleccionado de número entero del rango especificado.	<pre>import random print(random.randint(3, 9)) #devuelve un número entre 3 y 9 (ambos incluidos)</pre>
El método uniform() devuelve un número flotante aleatorio entre los dos números especificados (ambos incluidos).	<pre>import random print(random.uniform(20, 60))</pre>
El método choice() devuelve un elemento seleccionado aleatoriamente de la secuencia especificada.	<pre>import random x = "saludo" print(random.choice(x))</pre>

Ejemplos

Obtener 3 números con muchos decimales entre 0 y 1:

```
import random
for numero in range(3):
    print(random.random(), end=' ')
```

Obtener diez números enteros entre los que pueden aparecer números negativos, positivos y el cero:

```
import random
for numero in range(10):
    print(random.randint(-3, 3))
```

Obtener 25 números que van desde el 3 al 16 (no considera el 16):

```
import random
print('Valores posibles: 3, 6, 9, 12, 15')
for i in range(25):
    print(random.randrange(3, 16, 3), end=' ')
```

Obtener 3 números flotantes aleatorios entre 100 y 105:

```
import random
for numero in range(3):
    print(random.uniform(100, 105), end=' ')
```

*Obtener un valor de la secuencia especificada:
(Una lista sea crea con [] separando sus elementos con coma)*

```
import random
random.choice([1, 3.1415, "Pi", True])
```


Escenario

Simular el lanzamiento de un dado, imprimir el número que arrojó y si es un número par o impar.

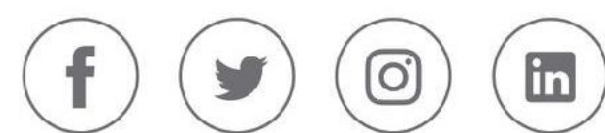
Salida propuesta por pantalla

```
El número que arrojó el dado fue: 1  
Este es un número impar.
```

```
El número que arrojó el dado fue: 6  
Este es un número par.
```

Otros módulos

<pre>import os</pre> <p>El módulo os nos permite acceder a funcionalidades dependientes del Sistema Operativo.</p>	<pre>import os os.system('cls') #genera un limpiado de pantalla</pre>
<pre>import time</pre> <p>Este módulo proporciona varias funciones relacionadas con el tiempo.</p>	<pre>import time time.sleep(15) #espera en segundos</pre>
<pre>import msvcrt</pre> <p>Este módulo permite la ejecución de rutinas útiles del entorno de ejecución.</p>	<pre>import msvcrt msvcrt.getch() #lee una pulsación de una tecla</pre>



inacap.cl