

### 1.1.1.11 RESUMEN DE SECCIÓN.

```
'''
    Autor: Christian Geovanni Arredondo Rangel
    Fecha: 23/Noviembre/2023
    Descrpcion: 1.1.1.11 RESUMEN DE SECCIÓN.
'''

#Ejercicio 1
'''
Quieres invocar la función make_money() contenida en el módulo
llamado mint. Tu código comienza con la siguiente línea:
'''
import mint
#¿Cuál es la forma adecuada de invocar a la función?
mint.make_money()

#Ejercicio 2
'''
Quieres invocar la función make_money() contenida en el módulo llamado mint.
Tu código comienza con la siguiente línea:
'''
from mint import make_money
#¿Cuál es la forma adecuada de invocar a la función?
make_money()

'''

'''
    Autor: Christian Geovanni Arredondo Rangel
    Fecha: 23/Noviembre/2023
    Descrpcion: 1.1.1.11 RESUMEN DE SECCIÓN.
'''

#Ejercicio 3
'''
Has escrito una función llamada make_money por tu cuenta. Necesitas importar una función con el mismo nombre
del módulo mint y no deseas cambiar el nombre de ninguno de tus nombres previamente definidos. ¿Qué variante
de la sentencia import puede ayudarte con el problema?
'''
# solución de muestra
from mint import make_money as make_more_money

#Ejercicio 4
#¿Qué forma de invocación de la función make_money es válida si tu código comienza con la siguiente línea?
from mint import *

make_money()
```

### 1.2.1.17 RESUMEN DE LA SECCIÓN

```
1.2.1.17 RESUMEN DE SECCIÓN.py - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
1.2.1.17 RESUMEN DE SECCIÓN.py X
home > blogh > Programacion-de-redes-2023 > R1 > Actividad2 > 1.2.1.17 RESUMEN DE SECCIÓN.py > ...
4 Descrpcion: 1.2.1.17 RESUMEN DE SECCIÓN.
5
6 '''
7 #Ejercicio 1
8 #¿Cuál es el valor esperado de la variable result después de que se ejecuta el siguiente código?
9
10 import math
11 result = math.e == math.exp(1)
12 print(result)
13
14 #Ejercicio 2
15 #(Completa el enunciado) Establecer la semilla del generador
16 # con el mismo valor cada vez que se ejecuta tu programa garantiza que ...
17
18 #... los valores pseudoaleatorios emitidos desde el módulo random serán exactamente los mismos.
19
20 #Ejercicio 3
21 #¿Cuál de las funciones del módulo platform utilizarías para determinar el nombre del CPU que corre dentro de tu computadora?
22
23 #La función processor()
24
25 #Ejercicio 4
26
27 #¿Cuál es el resultado esperado del siguiente fragmento de código?
28 import platform
29
30 print(len(platform.python_version_tuple()))
31
32 True
33
34 blogh@blogh:~/Programacion-de-redes-2023/R1/Actividad2$
```

### 1.3.1.11 RESUMEN DE LA SECCIÓN

```
1.3.1.11 RESUMEN DE LA SECCIÓN.py 7
home > blogh > Programacion-de-redes-2023 > R1 > Actividad2 > 1.3.1.11 RESUMEN DE LA SECCIÓN.py
1 '''
2 Autor: Christian Giovanni Arredondo Rangel
3 Fecha: 23/Noviembre/2023
4 Descrpcion: 1.3.1.11 RESUMEN DE LA SECCIÓN
5
6 '''
7 #Ejercicio 1
8 #Deseas evitar que el usuario de tu módulo ejecute tu código como un script ordinario. ¿Cómo lograrías tal efecto?
9 import sys
10
11 if __name__ == "__main__":
12     print ("¡No hagas eso!")
13     sys.exit()
```

```
1.3.1.11 RESUMEN DE LA SECCIÓN.py 7
home > blogh > Programacion-de-redes-2023 > R1 > Actividad2 > 1.3.1.11 RESUMEN DE LA SECCIÓN.py
1 '''
2 Autor: Christian Giovanni Arredondo Rangel
3 Fecha: 23/Noviembre/2023
4 Descrpcion: 1.3.1.11 RESUMEN DE LA SECCIÓN
5
6 '''
7
8 #Ejercicio 2
9 #Algunos paquetes adicionales y necesarios se almacenan dentro del directorio D:\Python\Project\Modules. Escribe
10 # un código asegurándote de que Python recorra el directorio para encontrar todos los módulos solicitados.
11 import sys
12
13 # ¡Toma en cuenta las diagonales invertidas dobles!
14 sys.path.append("D:\\Python\\Project\\Modules")
15
16 #Ejercicio 3
17 #El directorio mencionado en el ejercicio anterior contiene un subárbol con la siguiente estructura:
18
19 abc "abc" no está definido
20 | def sangSangría inesperadaria inesperada
21 | mymodule.py sangSangría inesperadaria inesperada
22
23
24 import abc.def.mymodule No se espera sangría
25
26 blogh@blogh:~/Programacion-de-redes-2023/R1/Actividad2$
```

#### 1.4.1.18 RESUMEN DE LA SECCIÓN

##### Ejercicio 1

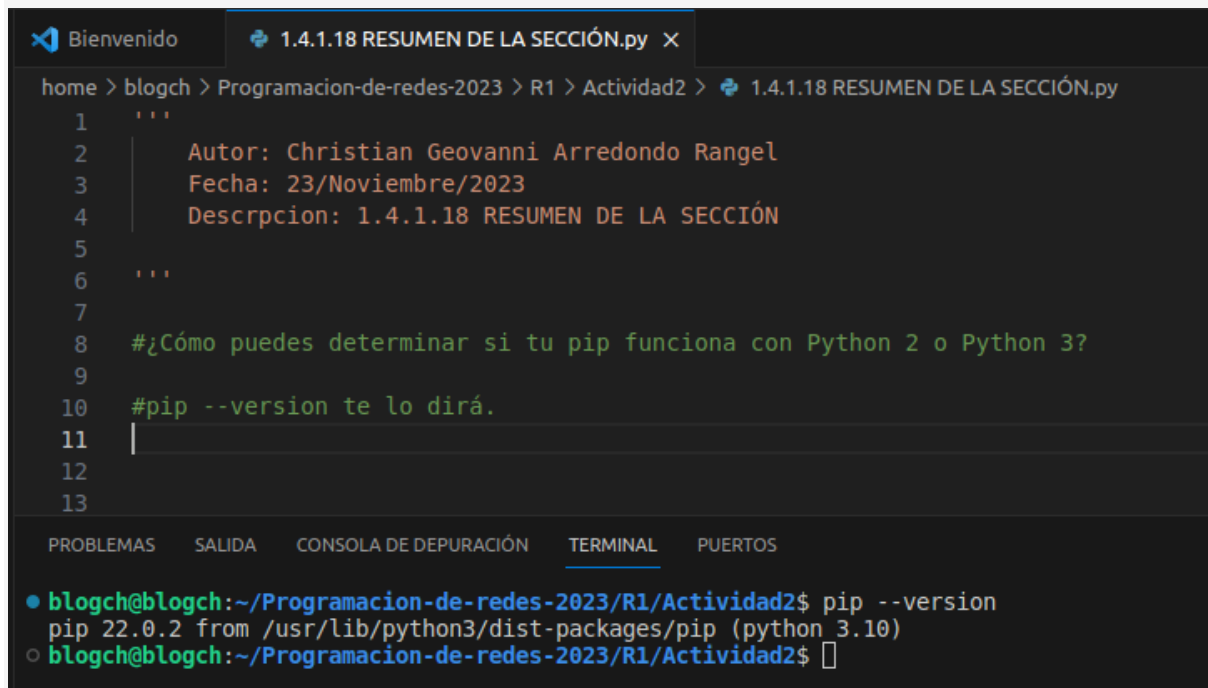
¿De donde proviene el nombre The Cheese Shop?

R=Es una referencia a un viejo sketch de *Monty Python* que lleva el mismo nombre.

##### Ejercicio 2

¿Por qué deberías asegurarte de cuál pip o pip3 es el correcto para ti?

R=Cuando Python 2 y Python 3 coexisten en el sistema operativo, es probable que pip identifique la instancia de pip que trabaja solo con paquetes de Python 2.



The screenshot shows a code editor with a file named '1.4.1.18 RESUMEN DE LA SECCIÓN.py'. The script contains a docstring with author, date, and description information, followed by a comment asking how to determine if pip works with Python 2 or 3, and a hint to use 'pip --version'. Below the code, the terminal output shows the command 'pip --version' being executed, resulting in 'pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)'.

```
Bienvenido 1.4.1.18 RESUMEN DE LA SECCIÓN.py x
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 1.4.1.18 RESUMEN DE LA SECCIÓN.py
1  '''
2      Autor: Christian Giovanni Arredondo Rangel
3      Fecha: 23/Noviembre/2023
4      Descrpion: 1.4.1.18 RESUMEN DE LA SECCIÓN
5
6  '''
7
8  #¿Cómo puedes determinar si tu pip funciona con Python 2 o Python 3?
9
10 #pip --version te lo dirá.
11 |
12
13
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
• blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ pip --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
○ blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$
```

##### Ejercicio 4

Desafortunadamente, no tienes privilegios de administrador. ¿Qué debes hacer para instalar un paquete en todo el sistema?

R=Tienes que consultar a tu administrador del sistema â “ ¡no intentes hackear tu sistema operativo!

#### 2.1.1.4 RESUMEN DE LA SECCIÓN

```

1  '''
2      Autor: Christian Geovanni Arredondo Rangel
3      Fecha: 23/Noviembre/2023
4      Descrpcion: 2.1.1.4 RESUMEN DE LA SECCIÓN
5
6
7  '''
8
9  Ejercico 1
10
11  ¿Qué es BOM?
12
13  BOM (Byte Order Mark), Una Marca de Orden de Bytes es una combinación
14  especial de bits que anuncia la codificación utilizada por el contenido
15  de un archivo (por ejemplo, UCS-4 o UTF-B).
16
17
18  Ejercico 2
19
20  ¿Está Python 3 internacionalizado?
21
22  Sí, está completamente internacionalizado: podemos usar caracteres
23  UNICODE dentro de nuestro código, leerlos desde la entrada y enviarlos a la salida.
24
25  '''

```

#### 2.2.1.15 RESUMEN DE LA SECCIÓN

The image shows a Visual Studio Code editor window with the title "2.2.1.15 RESUMEN DE LA SECCIÓN.py - Visual Studio Code". The editor is open to a file named "2.2.1.15 RESUMEN DE LA SECCIÓN.py". The code in the editor is as follows:

```
1  """
2  Autor: Christian Giovanni Arredondo Rangel
3  Fecha: 23/Noviembre/2023
4  Descrpcion: 2.2.1.15 RESUMEN DE LA SECCIÓN
5  """
6  #Ejercicio 1
7  #¿Cuál es la longitud de la siguiente cadena asumiendo que no hay espacios en blanco entre las comillas?
8  """
9  """
10 print("la longitud de la siguiente cadena es de 1")
11
12 #Ejercicio 2
13 #¿Cuál es el resultado esperado del siguiente código?
14 s = 'yesteryears'
15 the_list = list(s)
16 print(the_list[3:6])
17
18 #Ejercicio 3
19 #¿Cuál es el resultado esperado del siguiente código?
20
21 for ch in "abc":
22     print(chr(ord(ch) + 1), end='')
23
24
```

The terminal at the bottom shows the command to run the script and its output:

```
blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ /bin/python3 "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/2.2.1.15 RESUMEN DE LA SECCIÓN.py"
la longitud de la siguiente cadena es de 1
['t', 'e', 'r', 's']
bcdblogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$
```

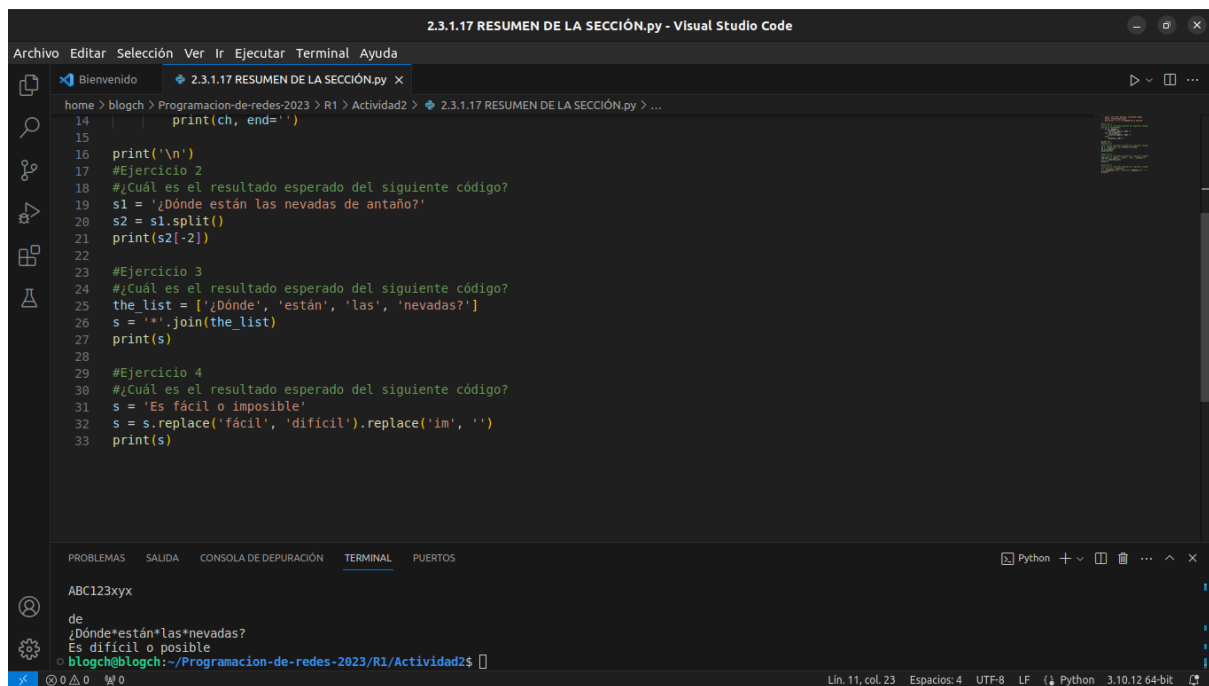
### 2.3.1.17 RESUMEN DE LA SECCIÓN

## Ejercicio 1

```
1 '''
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descrpcion: 2.3.1.17 RESUMEN DE LA SECCIÓN
5 '''
6 #Ejercicio 1
7 #¿Cuál es el resultado esperado del siguiente código?
8 for ch in "abc123XYX":
9     if ch.isupper():
10        print(ch.lower(), end='')
11    elif ch.islower():
12        print(ch.upper(), end='')
13    else:
14        print(ch, end='')

Console >_

ABC123xyx
```



## 2.4.1.5 RESUMEN DE LA SECCIÓN

```
2.4.1.5 RESUMEN DE LA SECCIÓN.py - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
x Bienvenido 2.4.1.5 RESUMEN DE LA SECCIÓN.py x
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 2.4.1.5 RESUMEN DE LA SECCIÓN.py > ...
6 #Ejercicio 1
7 #¿Cuál de las siguientes líneas describe una condición verdadera?
8 'smith' > 'Smith'
9 'Smiths' < 'Smith'
10 'Smith' > '1000'
11 '11' < '8'
12 print("El resultado de la actividad 1, es 1, 3 y 4")
13
14 #Ejercicio 2
15 #¿Cuál es el resultado esperado del siguiente código?
16 s1 = '¿Dónde están las nevadas de antaño?'
17 s2 = s1.split()
18 s3 = sorted(s2)
19 print(s3[1])
20
21 #Ejercicio 3
22 #¿Cuál es el resultado esperado del siguiente código?
23 s1 = '12.8'
24 i = int(s1)
25 s2 = str(i)
26 f = float(s2)
27 print([s1 == s2])

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + - [] ... ^ x
b:blogchblogch:~/Programacion-de-redes-2023/R1/Actividad2$ /bin/python3 "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/2.4.1.5 RESUMEN DE LA SECCIÓN.py"
El resultado de la actividad 1, es 1, 3 y 4
de
Traceback (most recent call last):
  File "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/2.4.1.5 RESUMEN DE LA SECCIÓN.py", line 24, in <module>
    i = int(s1)
ValueError: invalid literal for int() with base 10: '12.8'
b:blogchblogch:~/Programacion-de-redes-2023/R1/Actividad2$
```

## 2.5.1.5 RESUMEN DE LA SECCIÓN

Actividades Brave Web Browser 23 de nov 16:10

Course: Programación de Course: Programación de Edube Interactive :: 2. BlogCH (Christian Arredo) ChatGPT

edube.org/learn/python-essentials-2-esp/resumen-de-secci-oacute-n-20

2.5.1.5 RESUMEN DE SECCIÓN

MODULE (54%) SECTION (100%)

### Puntos Claves

1. Las cadenas son herramientas clave en el procesamiento de datos modernos, ya que la mayoría de los datos útiles son en realidad cadenas. Por ejemplo, el uso de un motor de búsqueda web (que parece bastante trivial en estos días) utiliza un procesamiento de cadenas extremadamente complejo, que involucra cantidades inimaginables de datos.
2. El comparar cadenas de forma estricta (como lo hace Python) puede ser muy insatisfactorio cuando se trata de búsquedas avanzadas (por ejemplo, durante consultas extensas a bases de datos). En respuesta a esta demanda, se han creado e implementado una serie de algoritmos de comparación de cadenas *difusos*. Estos algoritmos pueden encontrar cadenas que no son iguales en el sentido de Python, pero que son **similares**. Uno de esos conceptos es la **Distancia Hamming**, que se utiliza para determinar la similitud de dos cadenas. Si este tema te interesa, puedes encontrar más información al respecto aquí: [https://en.wikipedia.org/wiki/Hamming\\_distance](https://en.wikipedia.org/wiki/Hamming_distance). Otra solución del mismo tipo, pero basada en un supuesto diferente, es la **Distancia Levenshtein** descrita aquí: [https://en.wikipedia.org/wiki/Levenshtein\\_distance](https://en.wikipedia.org/wiki/Levenshtein_distance).
3. Otra forma de comparar cadenas es encontrar su similitud acústica, lo que significa un proceso que lleva a determinar si dos cadenas suenan similares (como "echo" y "hecho"). Esa similitud debe establecerse para cada idioma (o incluso dialecto) por separado. Un algoritmo utilizado para realizar una comparación de este tipo para el idioma Inglés se llama **Soundex** y se inventó, no lo creas, en 1918. Puedes encontrar más información al respecto aquí: <https://en.wikipedia.org/wiki/Soundex>.
4. Debido a la precisión limitada de los datos enteros y flotantes nativos, a veces es razonable almacenar y procesar valores numéricos enormes como cadenas. Esta es la técnica que usa Python cuando se le fuerza a operar con un número entero que consta de una gran cantidad de dígitos.

Prev Next

## 2.6.1.12 RESUMEN DE LA SECCIÓN

```
Bienvenido 2.6.1.12 RESUMEN DE LA SECCIÓN.py 1 X
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 2.6.1.12 RESUMEN DE LA SECCIÓN.py
1  '''
2  | Autor: Christian Geovanni Arredondo Rangel
3  | Fecha: 23/Noviembre/2023
4  | Descrpcion: 2.6.1.12 RESUMEN DE LA SECCIÓN
5  | '''
6  #Ejercicio 1
7  #¿Cuál es el resultado esperado del siguiente código?
8  try:
9  |     print("Tratemos de hacer esto")
10 |     print("#"[2])
11 |     print("¡Tuvimos éxito!")
12 except:
13 |     print("Hemos fallado")
14 print("Hemos terminado")
15
16 #Ejercicio 2
17 #¿Cuál es el resultado esperado del siguiente código?
18 try:
19 |     print("alpha"[1/0])
20 except ZeroDivisionError:
21 |     print("cero")
22 except IndexError: "IndexingError" no está definido
23 |     print("indice")
24 except:
25 |     print("algo")

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
Tratemos de hacer esto
Hemos fallado
Hemos terminado
cero
blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$
```

## 2.7.1.8 RESUMEN DE LA SECCIÓN

```
Bienvenido 2.7.1.8 RESUMEN DE LA SECCIÓN.py x
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 2.7.1.8 RESUMEN DE LA SECCIÓN.py > ...
1  '''
2      Autor: Christian Geovanni Arredondo Rangel
3      Fecha: 23/Noviembre/2023
4      Descrpcion: 2.7.1.8 RESUMEN DE LA SECCIÓN
5  '''
6  #Ejercicio 1
7  #¿Cuál es la salida esperada del siguiente código?
8  try:
9      print(1/0)
10 except ZeroDivisionError:
11     print("cero")
12 except ArithmeticError:
13     print("arit")
14 except:
15     print("algo")
16
17 #Ejercicio 2
18 #¿Cuál es la salida esperada del siguiente código?
19 try:
20     print(1/0)
21 except ArithmeticError:
22     print("arit")
23 except ZeroDivisionError:
24     print("cero")
25 except:
26     print("algo")
27

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

cero
arit
algo
o blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$
```

## 2.8.1.5 RESUMEN DE LA SECCIÓN



```
Bienvenido 2.8.1.5 RESUMEN DE LA SECCIÓN.py x
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 2.8.1.5 RESUMEN DE LA SECCIÓN.py > ...
1 '''
2     Autor: Christian Giovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descrpcion: 2.8.1.5 RESUMEN DE LA SECCIÓN
5 '''
6 #Ejercicio 1
7 #¿Cuál de las excepciones se utilizará para proteger al código de ser interrumpido por el uso
8 print(" la respuesta de la pregunta es KeyboardInterrupt")
9
10 #Ejercicio 2
11 #¿Cuál es el nombre de la más general de todas las excepciones de Python?
12 print(" la respuesta de la pregunta es BaseException")
13
14 #Ejercicio 3
15 #¿Cuál de las excepciones será generada a través de la siguiente evaluación fallida?
16 huge_value = 1E250 ** 2

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
algo
blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ /bin/python3 "/home/blogch/Programacion-de-rede
la respuesta de la pregunta es KeyboardInterrupt
la respuesta de la pregunta es BaseException
Traceback (most recent call last):
  File "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/2.8.1.5 RESUMEN DE LA SECCIÓN.py", line 16,
    huge_value = 1E250 ** 2
OverflowError: (34, 'Numerical result out of range')
blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$
```

### 3.1.1.8 RESUMEN DE SECCIÓN.

Actividades Brave Web Browser 23 de nov 16:45

Course: Programación de Edube Interactive :: 3. BlogCH (Christian Arredondo) ChatGPT

edube.org/learn/python-essentials-2-esp/resumen-de-secci-oacute-n-24

#### 3.1.1.8 RESUMEN DE SECCIÓN

MODULE (11%) SECTION (100%)

#### Puntos Clave

- Una **clase** es una idea (más o menos abstracta) que se puede utilizar para crear varias encarnaciones; una encarnación de este tipo se denomina **objeto**.
- Cuando una clase se deriva de otra clase, su relación se denomina **herencia**. La clase que deriva de la otra clase se denomina **subclase**. El segundo lado de esta relación se denomina **superclase**. Una forma de presentar dicha relación es en un **diagrama de herencia**, donde:
  - Las superclases siempre se presentan **encima** de sus subclases.
  - Las relaciones entre clases se muestran como flechas dirigidas **desde la subclase hacia su superclase**.
- Los objetos están equipados con:
  - Un **nombre** que los identifica y nos permite distinguirlos.
  - Un conjunto de **propiedades** (el conjunto puede estar vacío).
  - Un conjunto de **métodos** (también puede estar vacío).
- Para definir una clase de Python, se necesita usar la palabra clave reservada `class`. Por ejemplo:

```
class This_Is_A_Class:
    pass
```
- Para crear un objeto de la clase previamente definida, se necesita usar la clase como si fuera una función. Por ejemplo:

#### Ejercicio 1

Si asumimos que pitones, víboras y cobras son subclases de la misma superclase, ¿cómo la llamarías?

Revisar

Serpiente, reptil, vertebrado, animal: todas estas respuestas son aceptables.

#### Ejercicio 2

Intenta nombrar algunas subclases de la clase Pitón.

Revisar

Pitón india, Pitón de Roca Sfricana, Pitón Bola, Pitón Birmana: la lista es larga.

#### Ejercicio 3

¿Puedes usar la palabra "class" para darle nombre a alguna de tus clases?

Revisar

¡No, no puedes, `class` es una palabra clave reservada!

Prev Next

### 3.2.1.13 RESUMEN DE SECCIÓN.

```
3.2.1.13 RESUMEN DE SECCIÓN.py - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
home > blogh > Programacion-de-redes-2023 > R1 > Actividad2 > 3.2.1.13 RESUMEN DE SECCIÓN.py > ...
1 '''
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descrpion: 3.2.1.13 RESUMEN DE SECCIÓN.
5 '''
6 #Ejercicio 1
7 #Suponiendo que hay una clase llamada Snakes, escribe la primera línea de la declaración de clase Python, expresando el hecho de que
8 print("La respuesta es class Python(Snakes):")
9
10 #Ejercicio 2
11 #Algo falta en la siguiente declaración, ¿qué es?
12
13 class Snakes:
14     def __init__():
15         self.sound = 'Sssssss'
16
17 print("El constructor __init__() carece del parámetro obligatorio (deberíamos llamarlo self para cumplir con los estándares).")
18
19 #Ejercicio 3
20 #Modifica el código para garantizar que la propiedad venomous sea privada.
21 class Snakes:
22     def __init__(self):
23         self.venomous = True
24
25 El código debería verse como sigue:
26
27 class Snakes
28     def __init__(self):
29         self.__venomous = True")
30 '''
```

### 3.3.1.9 RESUMEN DE SECCIÓN.

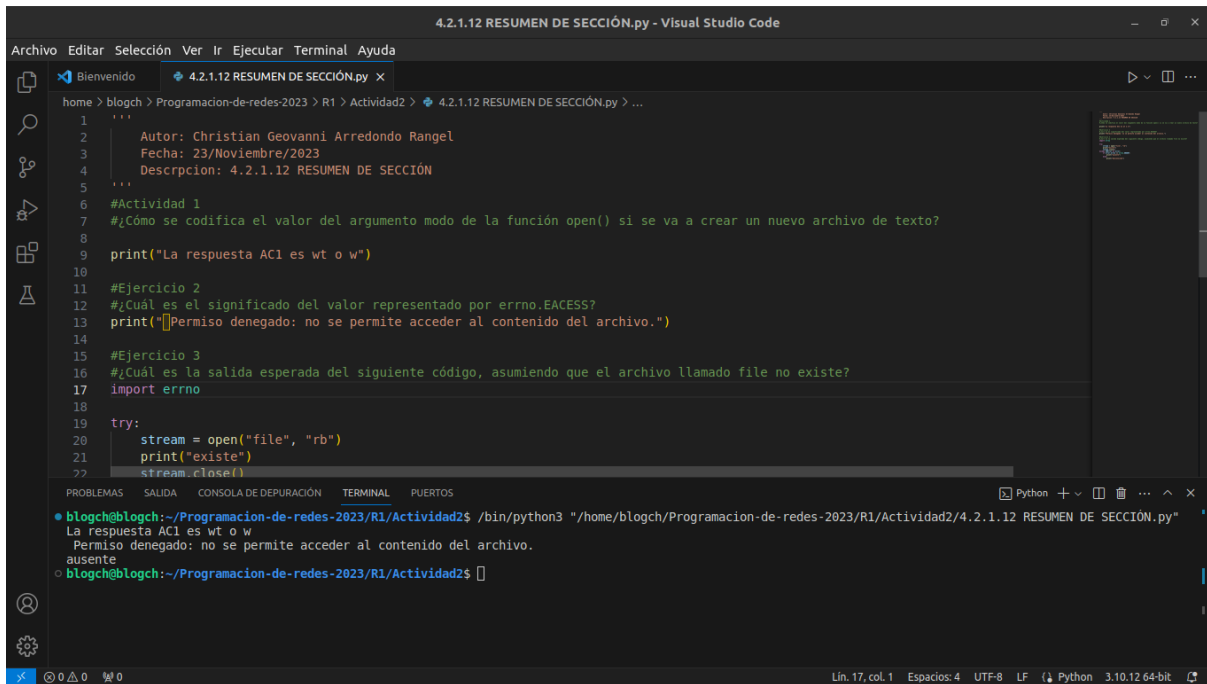
```
3.3.1.9 RESUMEN DE SECCIÓN.py - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
home > blogh > Programacion-de-redes-2023 > R1 > Actividad2 > 3.3.1.9 RESUMEN DE SECCIÓN.py > ...
1 '''
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descrpion: 3.3.1.9 RESUMEN DE SECCIÓN
5 '''
6 #Ejercicio 1
7 #¿Cuáles de las propiedades de la clase Python son variables de instancia y cuáles
8 # son variables de clase? ¿Cuáles de ellos son privados?
9 class Python:
10     population = 1
11     victims = 0
12     def __init__(self):
13         self.length_ft = 3
14         self.__venomous = False
15 #population y victims son variables de clase, mientras que length y __venomous son
16 # variables de instancia (esta última también es privada).
17
18 #Ejercicio 2
19 #Vas a negar la propiedad __venomous del objeto version_2, ignorando el hecho de que la propiedad es
20 # privada. ¿Cómo vas a hacer esto?
21 version_2 = Python()
22 version_2.__venomous = not version_2.__venomous
23
24 #Ejercicio 3
25 #Escribe una expresión que compruebe si el objeto version_2 contiene una propiedad de instancia
26 #denominada constrictor (¡sí, constrictor!).
27 hasattr(version_2, 'constrictor')
```

### 3.4.1.11 RESUMEN DE SECCIÓN.



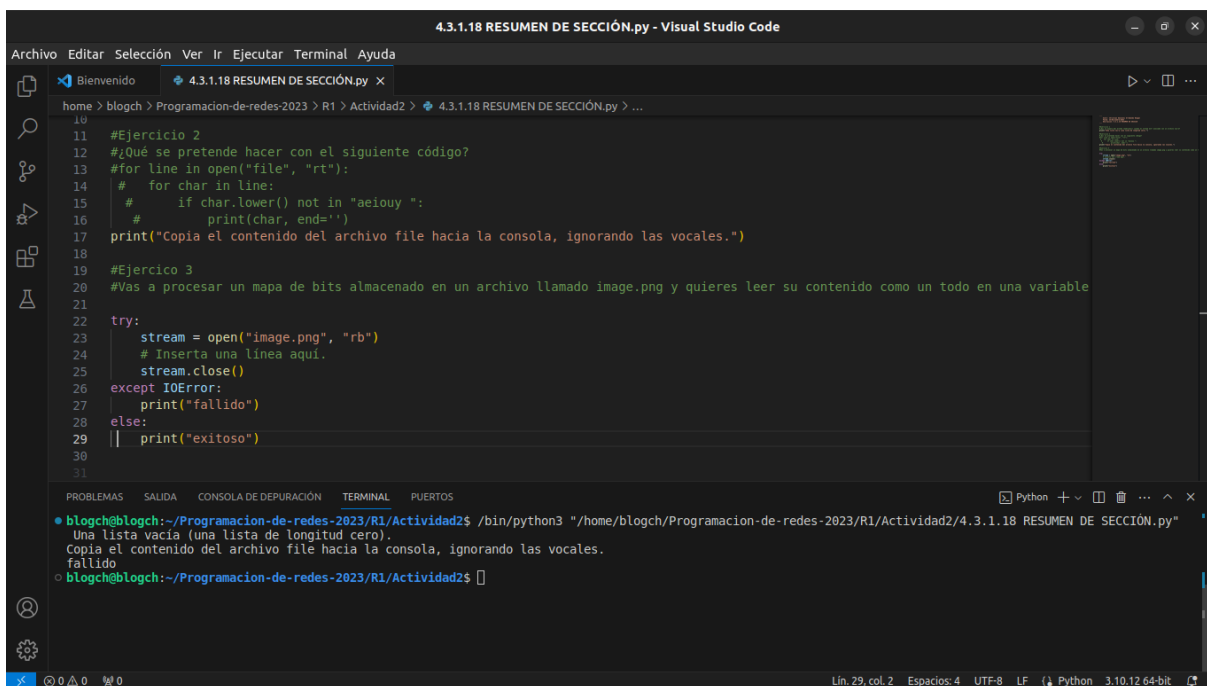






```
4.2.1.12 RESUMEN DE SECCIÓN.py - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
Bienvenido 4.2.1.12 RESUMEN DE SECCIÓN.py x
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 4.2.1.12 RESUMEN DE SECCIÓN.py > ...
1 '''
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descrpion: 4.2.1.12 RESUMEN DE SECCIÓN
5 '''
6 #Actividad 1
7 #¿Cómo se codifica el valor del argumento modo de la función open() si se va a crear un nuevo archivo de texto?
8
9 print("La respuesta AC1 es wt o w")
10
11 #Ejercicio 2
12 #¿Cuál es el significado del valor representado por errno.EACCESS?
13 print("[]Permiso denegado: no se permite acceder al contenido del archivo.")
14
15 #Ejercicio 3
16 #¿Cuál es la salida esperada del siguiente código, asumiendo que el archivo llamado file no existe?
17 import errno
18
19 try:
20     stream = open("file", "rb")
21     print("existe")
22     stream.close()
23
24 PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + - [] ... ^ x
• blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ /bin/python3 "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/4.2.1.12 RESUMEN DE SECCIÓN.py"
La respuesta AC1 es wt o w
[]Permiso denegado: no se permite acceder al contenido del archivo.
ausente
o blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ []
Lin. 17, col. 1 Espacios: 4 UTF-8 LF Python 3.10.12 64-bit
```

#### 4.3.1.18 RESUMEN DE SECCIÓN.



```
4.3.1.18 RESUMEN DE SECCIÓN.py - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
Bienvenido 4.3.1.18 RESUMEN DE SECCIÓN.py x
home > blogch > Programacion-de-redes-2023 > R1 > Actividad2 > 4.3.1.18 RESUMEN DE SECCIÓN.py > ...
10
11 #Ejercicio 2
12 #¿Qué se pretende hacer con el siguiente código?
13 for line in open("file", "rt"):
14     for char in line:
15         if char.lower() not in "aeiouy ":
16             print(char, end='')
17 print("Copia el contenido del archivo file hacia la consola, ignorando las vocales.")
18
19 #Ejercicio 3
20 #Vas a procesar un mapa de bits almacenado en un archivo llamado image.png y quieres leer su contenido como un todo en una variable
21
22 try:
23     stream = open("image.png", "rb")
24     # Inserta una línea aquí.
25     stream.close()
26 except IOError:
27     print("fallido")
28 else:
29     print("exitoso")
30
31 PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + - [] ... ^ x
• blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ /bin/python3 "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/4.3.1.18 RESUMEN DE SECCIÓN.py"
Una lista vacía (una lista de longitud cero).
Copia el contenido del archivo file hacia la consola, ignorando las vocales.
fallido
o blogch@blogch:~/Programacion-de-redes-2023/R1/Actividad2$ []
Lin. 29, col. 2 Espacios: 4 UTF-8 LF Python 3.10.12 64-bit
```

#### 4.4.1.9 RESUMEN DE SECCIÓN.

The screenshot shows the Visual Studio Code interface with a file named `4.4.1.9 RESUMEN DE SECCIÓN.py`. The code in the editor is as follows:

```
1
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descripción: 4.4.1.9 RESUMEN DE SECCIÓN
5     ...
6
7 #Ejercicio 1
8 #¿Cuál es el resultado del siguiente fragmento si se ejecuta en Unix?
9 import os
10 print(os.name)
11
12 #Ejercicio 2
13 #¿Cuál es el resultado del siguiente fragmento de código?
14 import os
15
16 os.mkdir("hello")
17 print(os.listdir())
18
```

The terminal at the bottom shows the command `/bin/python3 "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/4.4.1.9 RESUMEN DE SECCIÓN.py"` being executed. The output is a `Traceback (most recent call last):` error: `FileExistsError: [Errno 17] File exists: 'hello'`.

#### 4.5.1.23 RESUMEN DE SECCIÓN.

The screenshot shows the Visual Studio Code interface with a file named `4.5.1.23 RESUMEN DE SECCIÓN.py`. The code in the editor is as follows:

```
1
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descripción: 4.5.1.23 RESUMEN DE SECCIÓN
5     ...
6
7 #Ejercicio 1
8 #¿Cuál es el resultado del siguiente fragmento de código?
9 from datetime import time
10
11 t = time(14, 39)
12 print(t.strftime("%H:%M:%S"))
13
14 #Ejercicio 1
15 #¿Cuál es el resultado del siguiente fragmento de código?
16 from datetime import datetime
17
18 dt1 = datetime(2020, 9, 29, 14, 41, 0)
19 dt2 = datetime(2020, 9, 28, 14, 41, 0)
20 print(dt1 - dt2)
21
```

The terminal at the bottom shows the command `/bin/python3 "/home/blogch/Programacion-de-redes-2023/R1/Actividad2/4.5.1.23 RESUMEN DE SECCIÓN.py"` being executed. The output is `14:39:00` and `1 day, 0:00:00`.

#### 4.6.1.14 RESUMEN DE SECCIÓN.

4.6.1.14 RESUMEN DE SECCIÓN.py - Visual Studio Code

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

Bienvenido 4.6.1.14 RESUMEN DE SECCIÓN.py X

home > blogh > Programacion-de-redes-2023 > R1 > Actividad2 > 4.6.1.14 RESUMEN DE SECCIÓN.py > ...

```
1
2     Autor: Christian Geovanni Arredondo Rangel
3     Fecha: 23/Noviembre/2023
4     Descrpcion: 4.6.1.14 RESUMEN DE SECCIÓN
5     ...
6
7 #Ejercicio 1
8 #¿Cuál es el resultado del siguiente fragmento de código?
9 import calendar
10 print(calendar.weekheader(1))
11
12 #Ejercicio 2
13 #¿Cuál es el resultado del siguiente fragmento de código?
14 import calendar
15
16 c = calendar.Calendar()
17
18 for weekday in c.iterweekdays():
19     print(weekday, end=" ")
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + - [ ] ... ^ x

blogh@blogh:~/Programacion-de-redes-2023/R1/Actividad2\$ /bin/python3 "/home/blogh/Programacion-de-redes-2023/R1/Actividad2/4.6.1.14 RESUMEN DE SECCIÓN.py"

M T W T F S S

0 1 2 3 4 5 6 blogh@blogh:~/Programacion-de-redes-2023/R1/Actividad2\$

Lin. 12, col. 13 Espacios: 4 UTF-8 LF Python 3.10.12 64-bit