🎨

Lab 120: Secuencia de cadenas y peso numérico de la insulina

🔗Link: https://awsrestart.instructure.com/courses/1632/modules/items/886832

Información general sobre el laboratorio

En el módulo de conceptos básicos de Python, aprendió sobre variables, comentarios, cálculos, concatenaciones y excepciones. Ahora pondrá en práctica lo aprendido en la aplicación de la insulina en el mundo real.

Almacenará la secuencia de proteínas de la preproinsulina en una variable del tipo “string” y su peso en variables “int” y “float”. Luego, mostrará estas variables en la consola, con comentarios que expliquen el código. Realizará concatenaciones de textos y cálculos básicos.

En este laboratorio, deberá realizar lo siguiente:

- agregar comentarios que expliquen el propósito y el flujo del código
- utilizar `print()` para imprimir elementos del código Python en la consola
- utilizar manipulaciones de textos para obtener la secuencia de la insulina a partir de la preproinsulina
- realizar cálculos básicos sobre el peso molecular y la secuencia de la insulina
- asignar variables flotantes, enteras y de cadena de caracteres a números que representan el peso de la insulina
- explorar excepciones en Python

Ejercicio 1: Asignación de variables a los elementos de la secuencia de la insulina

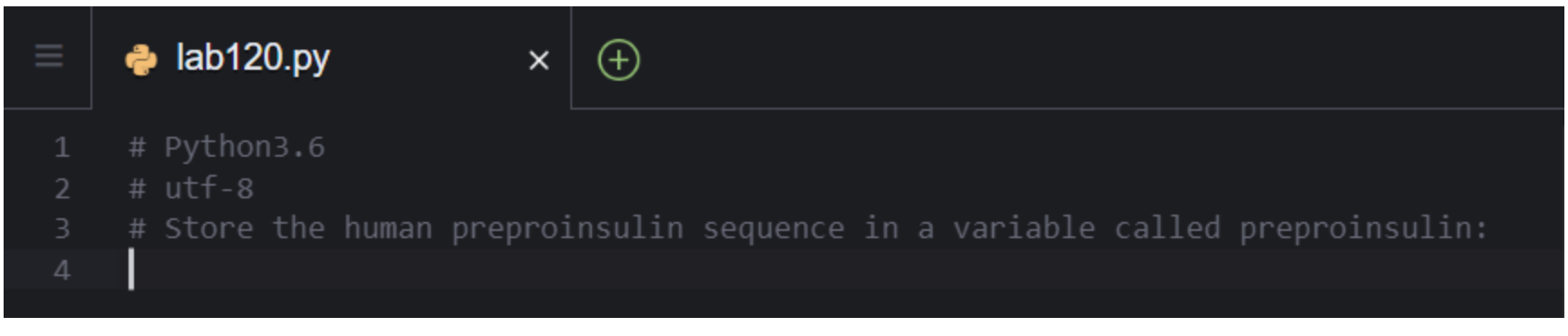
En este ejercicio, creará variables y les asignará un valor de tipo cadena.

1. En el panel de navegación del IDE, elija el archivo que creó en la sección *Creación del archivo de ejercicios de Python* anterior.

> Cómo empezar el archivo .py
>
> Siempre debe comenzar su archivo Python con comentarios. Recuerde que los comentarios de Python comienzan con un signo numeral (#).
>
> Los primeros comentarios deben proporcionar:
>
> - La versión de Python (*python3.6*) con una ruta al archivo ejecutable, si es posible
> - La codificación del archivo (normalmente, *coding: utf-8*)

2. Escriba este comentario en la siguiente línea:

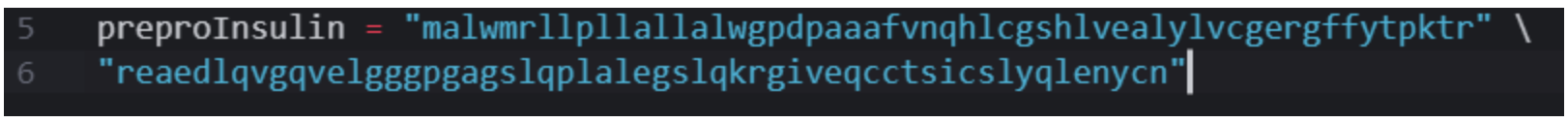
```
# Store the human preproinsulin sequence in a variable called preproinsulin:
```



3. Cree la primera variable en el archivo Python y escriba `preproInsulin =` como nombre de la variable y el signo igual (=) como el operador de asignación.
4. Después del signo igual (=), escriba la siguiente entrada:

```
"malwmrllpllallalwgpdpaaafvnqhlcgshlvealylvcgergffytpktr" \
"reaedlqvgqvelgggpgagslqplalegslqkrgiveqcctsicslyqlenycn"
```

5. Presione ENTER (Intro) para finalizar la primera variable en esa línea.

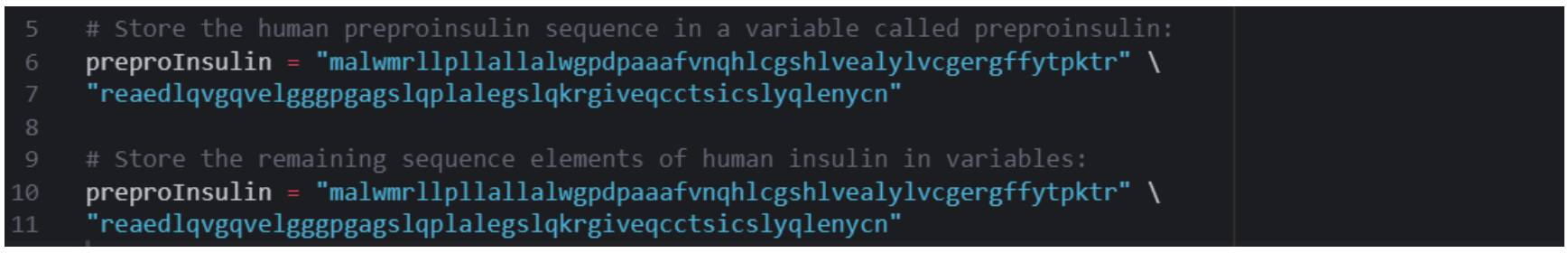


Longitud máxima de las líneas en archivos Python y otros estándares PEP

> 💡 La barra diagonal invertida final (\) en el valor de la variable del paso anterior se utiliza para mantener el cumplimiento de la guía de estilo Propuestas de Mejora de Python (PEP) 8. La guía de estilo PEP 8 recomienda un máximo de 79 caracteres por línea. Las PEP son estándares para las prácticas recomendadas de Python. Aunque el archivo también funciona con líneas más largas, respetar el límite sugerido aumenta los niveles de simplicidad y legibilidad. Mediante el uso de la barra invertida (\), puede dividir las variables y el código en bloques más pequeños y mantener el límite de 79 caracteres.

6. Escriba un comentario en el archivo:

```
# Store the remaining sequence elements of human insulin in variables:
```



7. Repita los pasos para definir una variable y asignarle un valor con la información del gráfico siguiente. Utilice un signo igual (=) entre el nombre de la variable y la cadena.

Nombre de la variable	Texto que se guardará en la variable
lsInsulin	"malwmrllpllallalwgpdpaaa"
bInsulin	"fvnqhlcgshlvealylvcgergffytpkt"
aInsulin	"giveqcctsicslyqlenycn"
cInsulin	"rreaedlqvgqvelgggpgagslqplalegslqkr"

> Nota: Los nombres de las variables en Python suelen comenzar con una primera palabra en minúsculas y luego, en mayúsculas para cada palabra siguiente, sin guiones bajos ni espacios. Mantenga la coherencia en la nomenclatura de las variables.

8. Por último, combinará los resultados de las agrupaciones de insulina más pequeñas en una única variable denominada *insulin*. Para hacer esto, en una nueva línea, ingrese: `insulin = bInsulin + aInsulin`.

Ejercicio 2: Uso de print() para mostrar secuencias de insulina en la consola

En este ejercicio, utilizará el método integrado `print()` para mostrar los elementos de la secuencia de la insulina en la consola.

1. Escriba un comentario en la línea siguiente:

```
# Printing "the sequence of human insulin" to console using successive print() commands:
```

2. En una nueva línea del archivo de Python, ingrese: `print("The sequence of human preproinsulin:")`.
3. Presione ENTER (Intro).

La instrucción `print()` imprime la representación directa de la cadena proporcionada, sin darle formato.

4. Para imprimir una cadena incluida en una variable del script, escriba lo siguiente: `print(preproInsulin)`.
5. Presione ENTER (Intro).

```
19  # Printing "the sequence of human insulin" to console using successive print() commands:
20  print("The sequence of human preproinsulin:")
21  print(preproInsulin)
```

6. Escriba el siguiente comentario:

```
# Printing to console using concatenated strings inside the print function (one-liner):
```

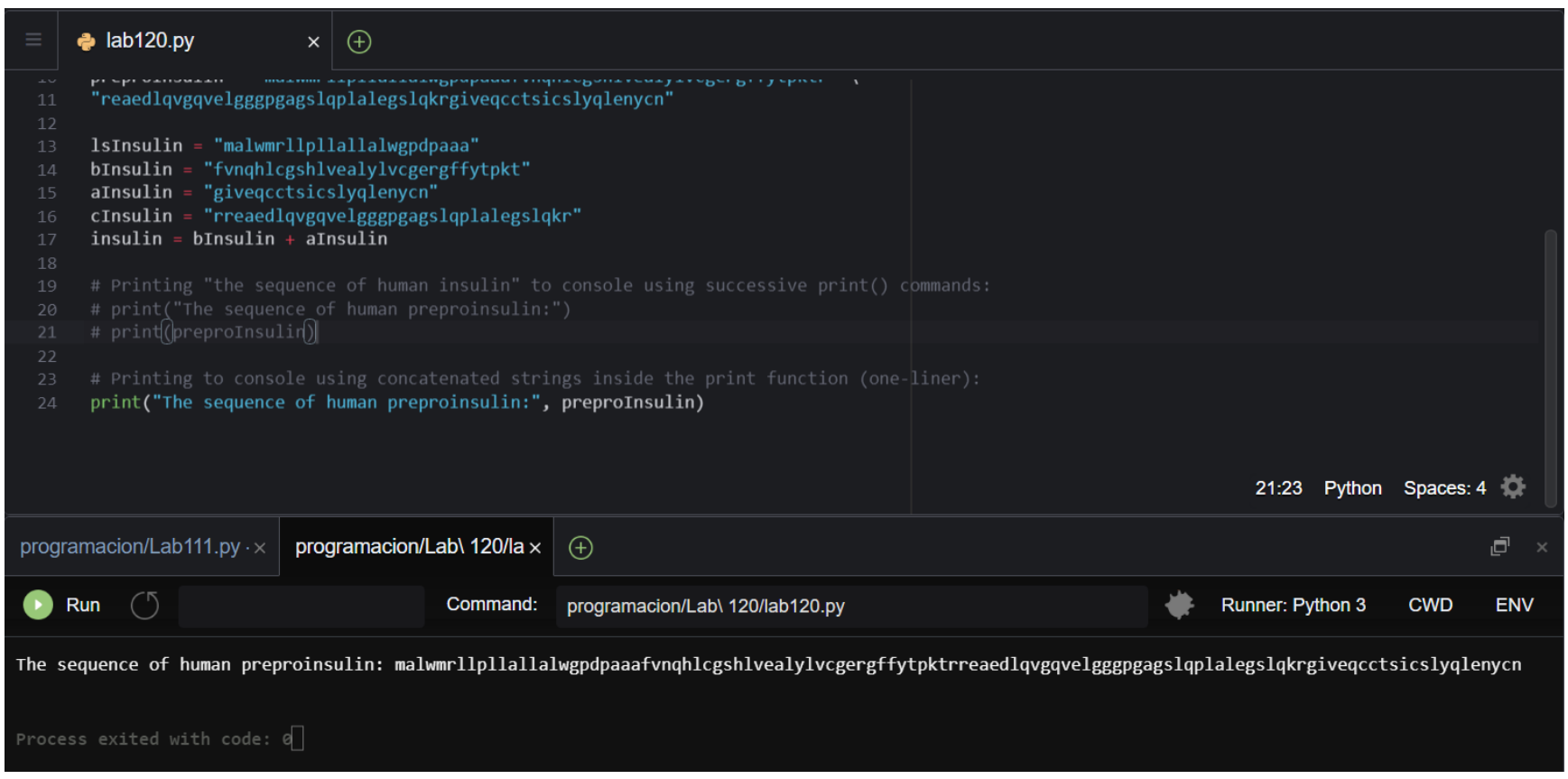
7. Para concatenar cadenas, utilice el signo más (+) en la instrucción `print()`:

```
print("The sequence of human insulin, chain a: " + aInsulin)
```

8. Presione ENTER (Intro).

> Nota: La función integrada print() acepta múltiples argumentos que pueden lograr la misma tarea en el paso 5. Por ejemplo: `print("The sequence of human insulin, chain a:", aInsulin)`

9. Guarde y ejecute el archivo.



Ejercicio 3: Cálculo del peso molecular aproximado de la insulina mediante el código proporcionado

En este laboratorio, calculará el peso molecular de la insulina, con el que trabajará en laboratorios posteriores.

1. Asegúrese de que el archivo **.py** esté abierto.
2. Copie el siguiente código y péguelo al final del archivo **.py**.

```
# Calculating the molecular weight of insulin
# Creating a list of the amino acid (AA) weights
aaWeights = {'A': 89.09, 'C': 121.16, 'D': 133.10, 'E': 147.13, 'F': 165.19,
```

```
'G': 75.07, 'H': 155.16, 'I': 131.17, 'K': 146.19, 'L': 131.17, 'M': 149.21,
'N': 132.12, 'P': 115.13, 'Q': 146.15, 'R': 174.20, 'S': 105.09, 'T': 119.12,
'V': 117.15, 'W': 204.23, 'Y': 181.19}
# Count the number of each amino acids
aaCountInsulin = ({x: float(insulin.upper().count(x)) for x in ['A', 'C',
'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T',
'V', 'W', 'Y']})
# Multiply the count by the weights
molecularWeightInsulin = sum({x: (aaCountInsulin[x]*aaWeights[x]) for x in
['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R',
'S', 'T', 'V', 'W', 'Y']}.values())
print("The rough molecular weight of insulin: " +
str(molecularWeightInsulin))
```

3. Guarde y ejecute el archivo.
4. Observe el resultado obtenido. Utilizará elementos de este código para trabajar con bucles y funciones en otros laboratorios, por lo tanto, observe cómo está escrito el código e intente seguir el resultado esperado.

> Nota: El peso molecular real de la insulina es de 5807,63, pero el programa muestra 6696,42 porque ignora ciertos enlaces y el procesamiento posterior a la traslación. Para calcular el porcentaje de error: error percentage = (| measured – accepted | / accepted)*100%.

5. Escriba o copie el ejemplo en el script.

```
molecularWeightInsulinActual = 5807.63
print("Error percentage: " + str(((molecularWeightInsulin - molecularWeightInsulinActual)/molecularWeightInsulinActual)*100))
```

6. Para ver el porcentaje de error, ejecute y guarde el archivo.

> Nota: Cuando utilice la concatenación de cadenas con cálculos con coma flotante, la función print() devuelve un error. Este error se gestiona con un método denominado casting, que indica a Python que debe utilizar un determinado tipo de datos. El uso previo de la función str() es un ejemplo de casting.

¡Felicitaciones! Ha trabajado con variables y diferentes tipos de datos en una función de Python.