



Módulo de autoaprendizaje N°20  
Tema: Variable y concatenación.

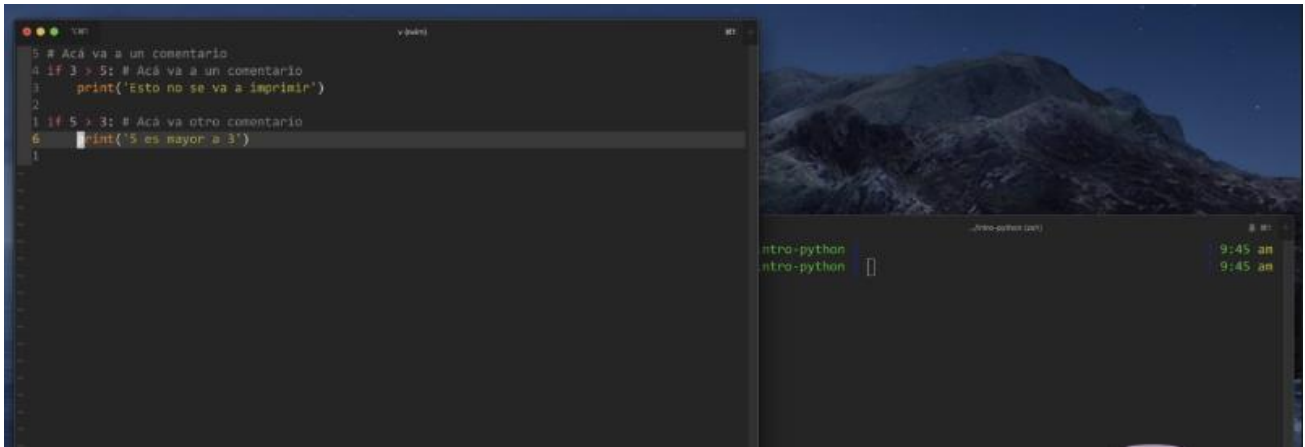
Objetivo: Comprender las múltiples variables y concatenación

Definición:

*Variables*

En este vídeo nosotros vamos a ver las variables. Una variable es un tipo de dato que en el fondo es como una cajita. Esta cajita va a contener valores que le vamos a asignar y después en un futuro podemos hacer referencias a estos valores.

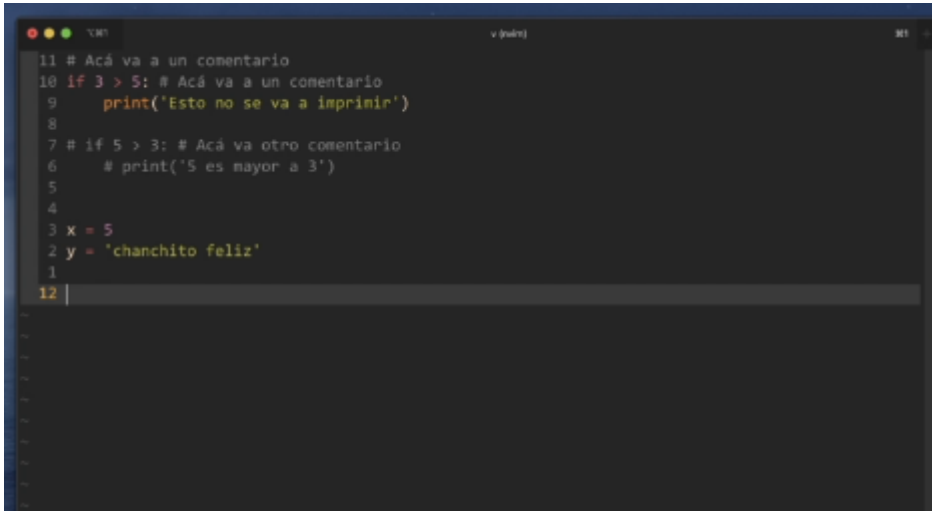
Entonces es como una cajita que contiene un tipo de dato. Continuamos utilizando esto mismo que se encuentra acá.



Y vamos a declarar dos variables. Vamos a declarar una variable X, que en este caso mi cajita de datos se llama X y estaba contener el valor de 5.

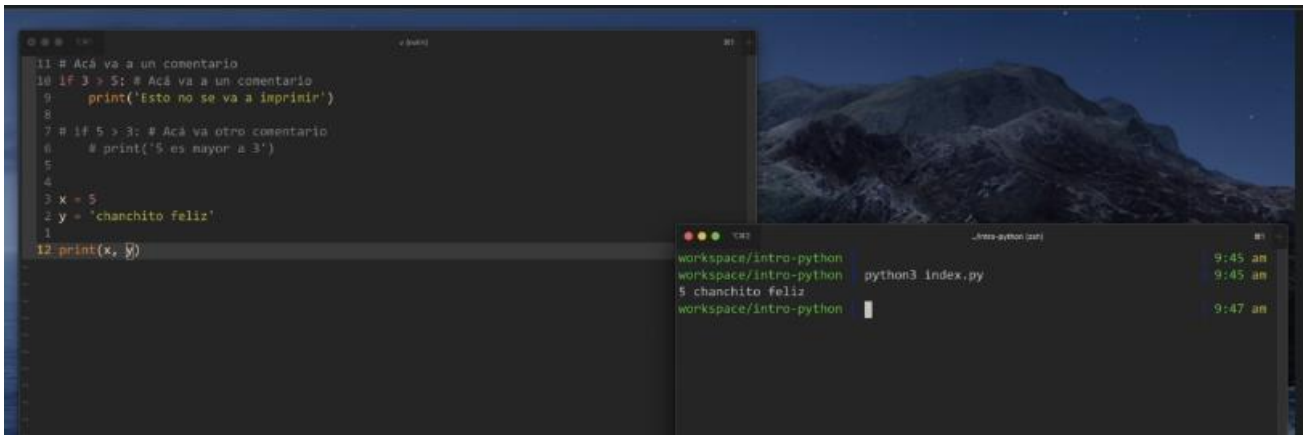
Voy a declarar otra variable que se va a llamar y y en este caso yo le voy a indicar un valor de un string que en este caso va a ser una palabra.

Y en este caso a este string yo le voy a colocar el nombre de "chanchito feliz".



Luego de esto, nosotros podemos hacer referencia a estos valores en un futuro.

Voy a escribir ahora print y le voy a pasar el valor de x separado por una coma el valor de y print me permite que yo le pueda pasar múltiples argumentos para después imprimirlos en la consola.



Cuando nosotros lo ejecutamos y como podemos ver, en este caso imprime el valor de 5, que en este caso es el valor que contenía nuestra variable X y también imprime chanchitos feliz que este era el valor que contenía nuestra variable y.



```
workspace/intro-python | python3 index.py | 9:45 am
workspace/intro-python | 9:45 am
5 chanchito feliz
workspace/intro-python | 9:47 am
```

De esta manera no es necesario que nosotros estemos trabajando constantemente con datos o valores primitivos.

Un valor primitivo siendo un número, siendo una palabra, una frase o una porción.

En este caso, nosotros podemos asignarle estos valores a variables y después, en un futuro hacemos referencia a éstas.

Continuando con este mismo hilo de las variables, nosotros le podemos asignar distintos nombres a nuestras variables.

En este caso, nosotros hemos utilizado x, e y, pero no es necesario que nosotros le coloquemos solamente letras.

Podemos utilizar, por supuesto, nombres que sean mucho más expresivos para que nosotros sepamos de qué se trata esta variable.

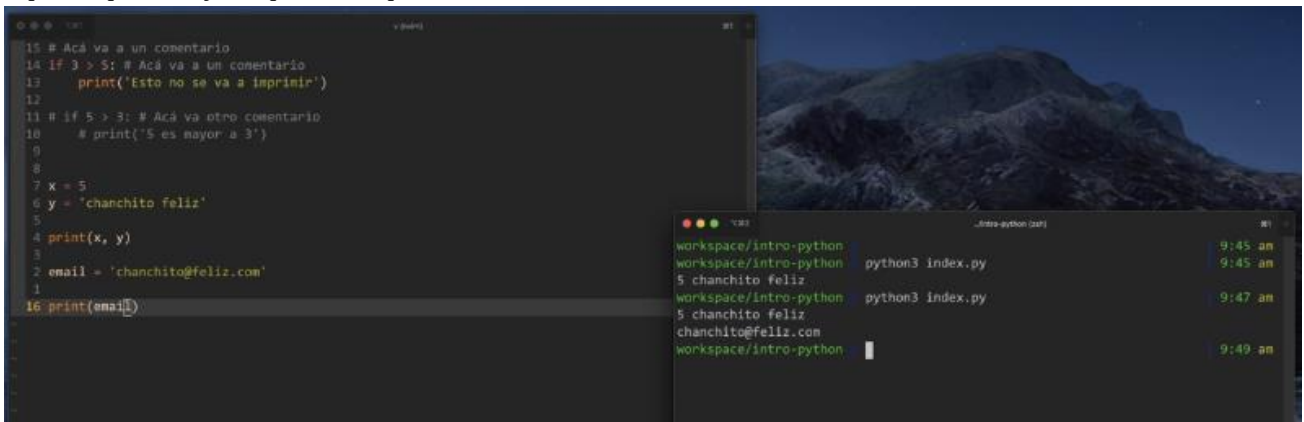
Vamos a continuar en esta misma línea, pero ahora vamos a crear una que se va a llamar email.

Y adivinen qué es lo que va a contener?

No, no va a contener un chanchito feliz, va a contener un correo electrónico.

Y en este caso voy a decir que el correo es el de “chanchitofeliz.com”.

Y por supuesto, yo la puedo imprimir en un futuro.



```
15 # Acá va a un comentario
14 if 3 > 5: # Acá va a un comentario
13     print('Esto no se va a imprimir')
12
11 # if 5 > 3: # Acá va otro comentario
10     # print('5 es mayor a 3')
9
8
7 x = 5
6 y = 'chanchito feliz'
5
4 print(x, y)
3
2 email = 'chanchitofeliz.com'
1
16 print(email)
```

```
workspace/intro-python | python3 index.py | 9:45 am
workspace/intro-python | 9:45 am
5 chanchito feliz
workspace/intro-python | python3 index.py | 9:47 am
5 chanchito feliz
chanchitofeliz.com
workspace/intro-python | 9:49 am
```

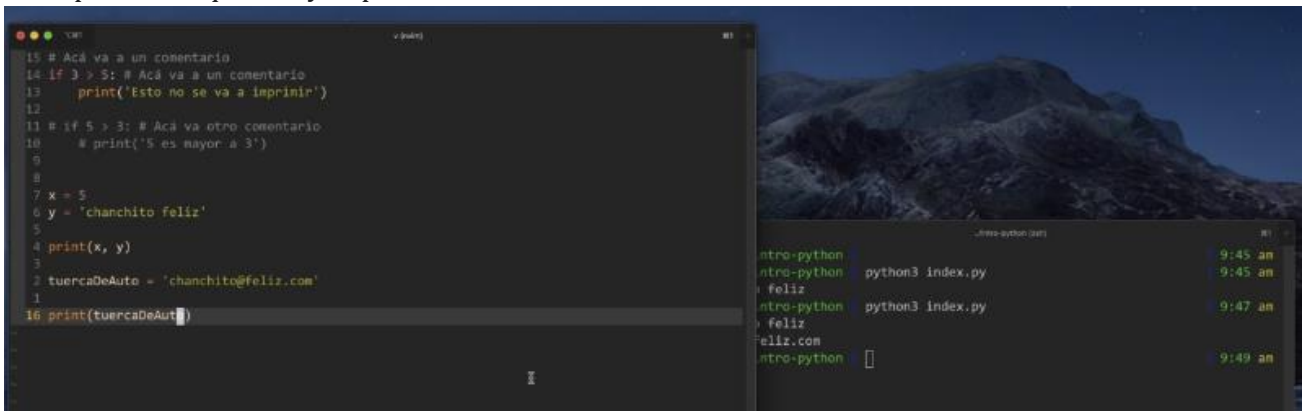
Vamos ahora a imprimir esto que se encuentra acá.

Y como pueden ver, seguimos imprimiendo el valor de 5 y también de chanchito feliz pero también estamos imprimiendo nuestro correo electrónico que se llama Chanchito feliz, o sea, chanchito@feliz.com.

Quiero que se fijen en esto, que es sumamente importante.

El nombre que yo le puedo dar a mis variables es completamente arbitrario.

O sea, yo en este caso podría decir que esto es una tuerca de auto y después cuando yo la imprima, voy a estar pensando que estoy imprimiendo una tuerca de auto.



```
15 # Acá va a un comentario
14 if 3 > 5: # Acá va a un comentario
13     print('Esto no se va a imprimir')
12
11 # if 5 > 3: # Acá va otro comentario
10     # print('5 es mayor a 3')
9
8
7 x = 5
6 y = 'chanchito feliz'
5
4 print(x, y)
3
2 tuercadeAuto = 'chanchitofeliz.com'
1
16 print(tuercadeAut)
```

```
ntro-python | python3 index.py | 9:45 am
ntro-python | 9:45 am
5 chanchito feliz
ntro-python | python3 index.py | 9:47 am
5 chanchito feliz
chanchitofeliz.com
ntro-python | 9:49 am
```

Pero no es una tuerca de auto, es un correo electrónico.

Y si yo lo imprimo, claro, veo chanchitos feliz no veo una tuerca de auto, por lo que hay que ser sumamente cuidadosos con los nombres que nosotros les vamos a dar a nuestras variables, de manera que éstas sean representativas del dato que ellos están guardando.

En este caso, como estamos guardando un correo electrónico lo mejor es que le demos el nombre de correo o email y no le demos otros nombres, como un nombre de usuario, edad o cosas así. Tenemos que tratar de ser consistentes y también de entregarle nombres representativos que hagan sentido.

Además de esto, nosotros podemos asignar distintos tipos de nombres a nuestras variables. Vamos a crear una nueva variable que se va a llamar “mi variable”. En este caso, mi variable se va a llamar “chanchito”.

```
17 # Acá va a un comentario
16 if 3 > 5: # Acá va a un comentario
15     print('Esto no se va a imprimir')
14
13 # if 5 > 3: # Acá va otro comentario
12     # print('5 es mayor a 3')
11
10
9 x = 5
8 y = 'chanchito feliz'
7
6 print(x, y)
5
4 correo = 'chanchito@feliz.com'
3
2 print(correo)
1
18 mi_var = 'chanchito'
```

Como pueden ver, este nombre de mi variable es un nombre válido.

Otra forma en la cual yo le puedo dar nombre a mis variables es utilizando un guión bajo para poder separar las distintas palabras que podría tener mi variable, como por ejemplo nombre usuario, puede ser una forma de darle nombre a esta variable y podemos utilizar el guión bajo para que se vea que es una palabra separada de la otra.

También nosotros podemos utilizar un guión bajo al comienzo para poder asignarle un nombre a mi variable y también puedo utilizar “camel case”. Camel case que es cuando nosotros, en lugar de colocar guiones bajos utilizamos la primera palabra la colocamos en mayúscula para poder separar una palabra de la otra y que ésta sea más fácil de leer.

Como en este caso estamos viendo “mi var”, también podemos colocar todo esto en mayúsculas.

Esto se utiliza por lo general para asignar constantes o variables de las cuales no puede cambiar su valor.

Y también nosotros podemos asignarle números a nuestras variables, pero siempre y cuando estos números se encuentren después de una letra.

```
18 # Acá va a un comentario
17 if 3 > 5: # Acá va a un comentario
16     print('Esto no se va a imprimir')
15
14 # if 5 > 3: # Acá va otro comentario
13     # print('5 es mayor a 3')
12
11
10 x = 5
9 y = 'chanchito feliz'
8
7 print(x, y)
6
5 correo = 'chanchito@feliz.com'
4
3 print(correo)
2
1 _mi_var = 'chanchito'
19 MIVAR23 = 'chanchito'
```

Nosotros no podemos asignar o crear variables que empiecen con números, como pueden ver aquí me lo está mostrando en rojo, indicando que estoy utilizando un nombre de variable no permitido, por lo

que necesariamente si le vamos a colocar números, estas van a tener que ser después de una letra.

```
18 # Acá va a un comentario
17 if 3 > 5: # Acá va a un comentario
16     print('Esto no se va a imprimir')
15
14 # if 5 > 3: # Acá va otro comentario
13     # print('5 es mayor a 3')
12
11
10 x = 5
9 y = 'chanchito feliz'
8
7 print(x, y)
6
5 correo = 'chanchito@feliz.com'
4
3 print(correo)
2
1 _mi_var = 'chanchito'
19 MIVAR = 'chanchito'
```

Tampoco puedo utilizar guiones dentro de mis variables para definirlas, tiene que ser o con un guión bajo o podemos hacer esa convención de utilizar letras mayúsculas y minúsculas para separar los distintos nombres de variables, o los distintos nombres que va a tener mi variable.

Y por supuesto, tampoco podemos utilizar un espacio para definir nuestras variables.

Todas tienen que estar juntas.

```
18 # Acá va a un comentario
17 if 3 > 5: # Acá va a un comentario
16     print('Esto no se va a imprimir')
15
14 # if 5 > 3: # Acá va otro comentario
13     # print('5 es mayor a 3')
12
11
10 x = 5
9 y = 'chanchito feliz'
8
7 print(x, y)
6
5 correo = 'chanchito@feliz.com'
4
3 print(correo)
2
1 _mi_var = 'chanchito'
19 MI VAR = 'chanchito'
```

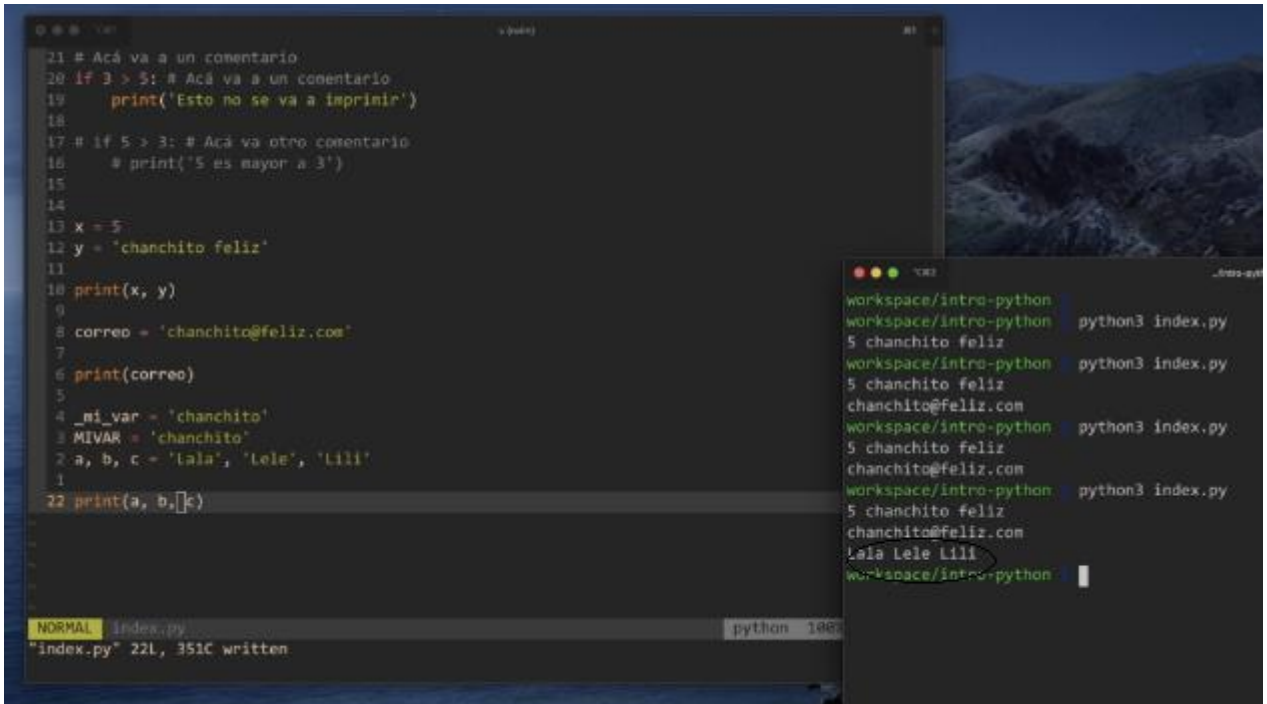
### Multiples variables

Python a nosotros también nos permite poder definir múltiples variables en una sola línea.

En este caso yo lo que puedo hacer es definir una variable A, B y C y que éstas tengan los valores de "Lala", "Lele" y "Lili", y luego, cuando yo quiera imprimir estos valores, voy a tener "A, B y C", voy a darles un pequeño espacio.

```
21 # Acá va a un comentario
20 if 3 > 5: # Acá va a un comentario
19     print('Esto no se va a imprimir')
18
17 # if 5 > 3: # Acá va otro comentario
16     # print('5 es mayor a 3')
15
14
13 x = 5
12 y = 'chanchito feliz'
11
10 print(x, y)
9
8 correo = 'chanchito@feliz.com'
7
6 print(correo)
5
4 _mi_var = 'chanchito'
3 MIVAR = 'chanchito'
2 a, b, c = 'Lala', 'Lele', 'Lili'
1
22 pr
print [A]
```

Y ahora, cuando yo las imprima, voy a ver qué A, B y C tienen los valores de Lala, Lele y Lili.



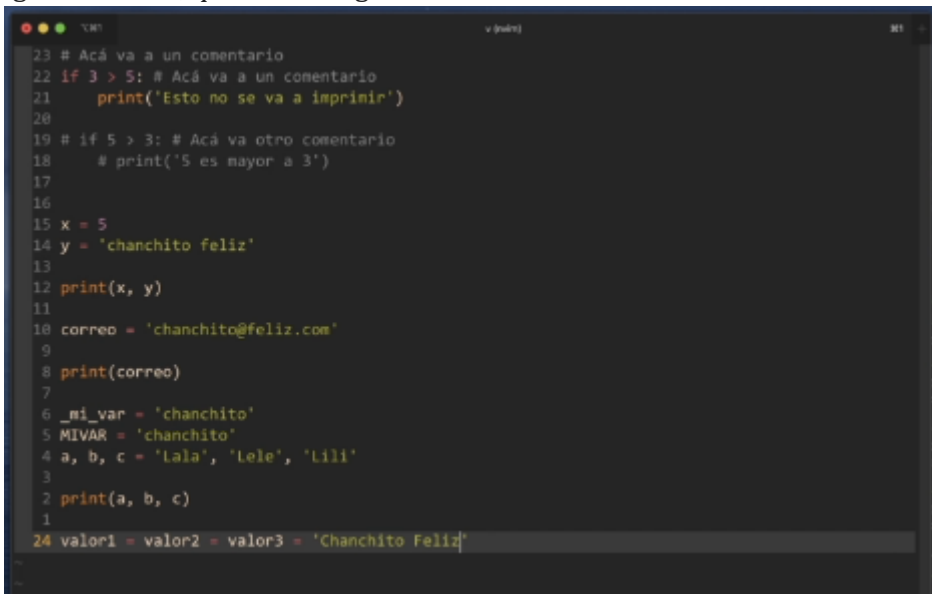
```
21 # Acá va a un comentario
20 if 3 > 5: # Acá va a un comentario
19     print('Esto no se va a imprimir')
18
17 # if 5 > 3: # Acá va otro comentario
16     # print('5 es mayor a 3')
15
14
13 x = 5
12 y = 'chanchito feliz'
11
10 print(x, y)
9
8 correo = 'chanchito@feliz.com'
7
6 print(correo)
5
4 _mi_var = 'chanchito'
3 MIVAR = 'chanchito'
2 a, b, c = 'Lala', 'Lele', 'Lili'
1
22 print(a, b, c)
```

```
workspace/intro-python python3 index.py
5 chanchito feliz
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
Lala Lele Lili
workspace/intro-python
```

NORMAL index.py python 186  
"index.py" 22L, 351C written

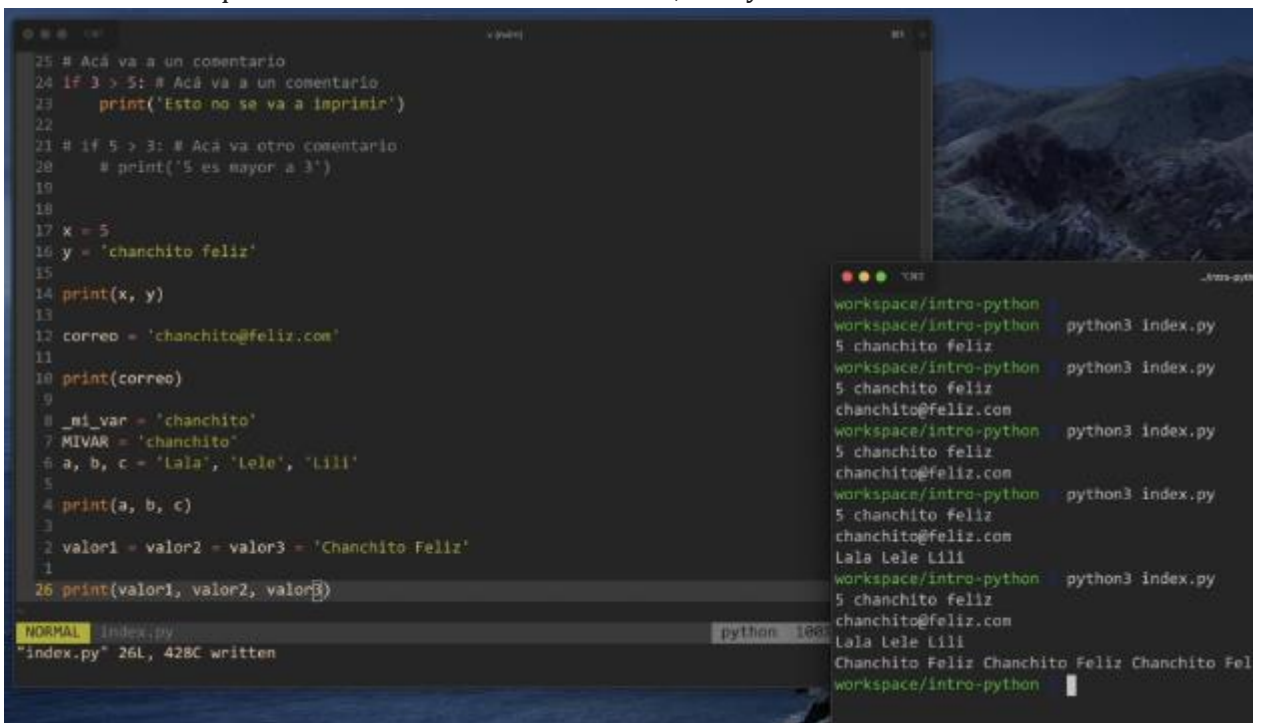
De esta manera nosotros podemos crear múltiples variables utilizando solamente una línea y en el caso que nosotros queramos definir muchas variables, pero éstas todas contengan el mismo valor, nosotros lo podemos hacer de una forma bastante similar.

Vamos a, en lugar de reutilizar a,b y c. Vamos a utilizar valor 1 va a ser igual a valor 2 y esta va a ser igual a valor 3, que va a ser igual a “chanchito feliz”.



```
23 # Acá va a un comentario
22 if 3 > 5: # Acá va a un comentario
21     print('Esto no se va a imprimir')
20
19 # if 5 > 3: # Acá va otro comentario
18     # print('5 es mayor a 3')
17
16
15 x = 5
14 y = 'chanchito feliz'
13
12 print(x, y)
11
10 correo = 'chanchito@feliz.com'
9
8 print(correo)
7
6 _mi_var = 'chanchito'
5 MIVAR = 'chanchito'
4 a, b, c = 'Lala', 'Lele', 'Lili'
3
2 print(a, b, c)
1
24 valor1 = valor2 = valor3 = 'Chanchito Feliz'
```

Y después, cuando yo quiera imprimir valor 1, valor 2 y valor 3, esta me va a mostrar el mismo valor constantemente que es el de chanchito feliz chanchito, feliz y chanchito feliz.



```
25 # Acá va a un comentario
24 if 3 > 5: # Acá va a un comentario
23     print('Esto no se va a imprimir')
22
21 # if 5 > 3: # Acá va otro comentario
20     # print('5 es mayor a 3')
19
18
17 x = 5
16 y = 'chanchito feliz'
15
14 print(x, y)
13
12 correo = 'chanchito@feliz.com'
11
10 print(correo)
9
8 _mi_var = 'chanchito'
7 MIVAR = 'chanchito'
6 a, b, c = 'Lala', 'Lele', 'Lili'
5
4 print(a, b, c)
3
2 valor1 = valor2 = valor3 = 'Chanchito Feliz'
1
26 print(valor1, valor2, valor3)
```

```
workspace/intro-python python3 index.py
5 chanchito feliz
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
Lala Lele Lili
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
Lala Lele Lili
Chanchito Feliz Chanchito Feliz Chanchito Fel
workspace/intro-python
```

NORMAL index.py python 186  
"index.py" 26L, 428C written

Y como pueden ver aquí nosotros tenemos chanchito feliz, chanchito feliz y más chanchitos felices. Son 3 chanchitos felices.

### Concatenación

Además de todo lo que ya hemos visto hasta ahora para python, también nos permite a nosotros realizar concatenación de palabras.

En este caso la concatenación es cuando yo tomo en este caso una palabra, tengo otra palabra y después yo las junto para que sea solamente una variable.

Vamos a definir en este caso otra variable que se va a llamar "inicio" y le vamos a dar el valor de "Hola!" Después de eso vamos a crear otra que se va a llamar "final" y vamos a asignarle en este caso el valor de "mundo".

```
3 inicio = 'Hola '
2 final = 'mundo'
1 print [A]
```

Ahora, cuando nosotros vayamos a imprimirlo, en el caso que queramos que estas dos variables se junten en una sola. lo que nosotros podemos hacer es utilizar la concatenación.

Nosotros tenemos que nombrar primero la variable que nosotros queremos concatenar al comienzo, seguido del símbolo más o en este caso símbolo de suma. Después de eso, tenemos que indicarle cuál es la otra variable que nosotros queremos concatenar.

En este caso voy a indicar que es final.

```
31 print(inicio + final)
```

De esta manera yo estoy concatenado la variable de inicio junto a la variable de final y el resultado que me va a entregar esto va a ser "Hola Mundo"

```
Hola mundo
```

Y es porque aquí yo tengo un pequeño espacio.

Vamos a ver qué es lo que imprime y como pueden ver aquí, yo tengo la frase de Hola mundo.

Esto es distinto a que nosotros utilicemos la coma .

Cuando nosotros utilizamos la coma me va a mostrar dos espacios, porque en el fondo está imprimiendola primera variable que es o con el espacio, luego separa esa variable de la que viene agregando el espacio y luego imprime la siguiente variable.

```
31 print(inicio, final)
NORMAL index.py
"index.py" 31L, 484C written
python 180
workspace/intro-python python3 index.py
5 chanchito feliz
chanchito@feliz.com
Lala Lele lili
Chanchito Feliz Chanchito Feliz Chanchito Fel
Hola mundo
workspace/intro-python
```

Ahora, cuando yo quiera imprimir en este caso la variable de inicio concatenadas con la variable de final, voy a ver que Hola mundo, se encuentra todo pegado.

Por lo que la concatenación nos va a servir para que nosotros podamos juntar en este caso palabras, que es lo que nosotros estamos haciendo acá.

1.- Ahora hazlo tú.

I. Ahora que revisaste el paso a paso te invito a ingreses a la aplicación para que puedas ejercitar ya que el próximo modulo comenzaremos los ejercicios.

2.- Revisa los resultados obtenidos

3.- Finalmente responde esta autoevaluación marcando la opción que corresponda luego de haber revisado tus respuestas.

Indicador	Sí	No
¿Revisé el paso a paso?		
¿Realicé el paso a paso?		