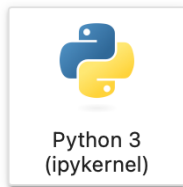


Conceptos Básicos de Programación

1-Correr jupyterlab en el ambiente creado - verificar que el directorio adecuado esta en uso

```
conda --version  
cd  
conda activate DrillingAnalytics  
jupyter-lab
```

2-Arrancar un Notebook Python 3



3- Escribir instrucción que imprima: "Hola"

```
print("Hola")
```

4- Calcular 2+2

```
2+2
```

5- Declarar dos variables llamadas a y b, con valores 2 y 3 respectivamente, y sumarlas

```
a=2  
b=3  
a+b
```

6- Intercambiando lost valores de variables - forma clásica

```
a=10  
b=30  
  
temp=a  
a=b  
b=temp  
print(a,b)
```

7- Intercambiando lost valores de variables - forma pythónica

```
a=10  
b=30  
  
a,b = b,a  
print(a,b)
```

8- Asignando variables valores por referencia

```
a=10
b=30
c=a
print(a,b,c)
```

```
a=20
print(a,b,c)
```

9- Declarar un array con valores 1,2,3,4,5 e imprimir el tercer elemento

```
c=[1,2,3,4,5]
print(c[2])
```

10- Declarar una lista con valores 1,2,3,4,"manzana","limón" e imprimir el quinto elemento

```
d=[1,2,3,4,"manzana","limon"]
print(d[4])
```

11. Agregar el elemento "naranja" al final de la lista e imprimir la lista completa

```
d.append("naranja")
print(d)
```

12. Agregar el valor de la variable a la lista e imprimir la lista completa

```
d.append(a)
print(d)
```

13. Agregar la variable (no el valor) a la lista e imprimir la lista completa

```
a=30
print(d)

a=10
a=[a]
d.append(a)
print(d)
```

14. Cambiar el valor de la variable a 30 e imprimir la lista d

```
a[0]=30
print(d)
```

15. Re-examinando la asignación de valores a variables, asignación por referencia. Este es un concepto pythónico no presentado en muchos otros lenguajes.

```
a=[10]
b=[30]
c=a
print(a,b,c)
```

```
a[0]=20
print(a,b,c)
```

16*. Ejercicio difícil: crear una matriz con datos the Profundidad, ROP, WOB y RPM. Para ser completado al final del día.

Este ejercicio introduce muchos conceptos nuevos, incluyendo: librerías, Loops y diccionarios

```
import random

random.seed(42)

sample_data = [
    {
        'depth': 500 + i,
        'ROP': random.uniform(20, 40),
        'WOB': random.uniform(2000, 2500),
        'RPM': random.uniform(100, 120)
    }
    for i in range(10)
]

print("Sample Data:")
for row in sample_data:
    print(row)
```

17. Código para iterar é imprimir los distintos elementos the una lista

```
parametros = ['ROP', 'WOB', 'RPM']
for parametro in parametros:
    print(parametro)
```

18. Código para desempacar variables

```
seq = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]
print(seq)

for a,b,c in seq:
    print(a,b,c)
```

19. Código para iterar é imprimir los distintos elementos the una lista incluyendo el indice

```
parametros = ['ROP', 'WOB', 'RPM']
for index, parametro in enumerate(parametros):
    print(f"Parametro at index {index}: {parametro}")
```

Como se podría obtener el mismo resultado sin utilizar *enumerate*?

```
parametros = ['ROP', 'WOB', 'RPM']
i=0
for parametro in parametros:
    print(f"Parametro at index {i}: {parametro}")
```

```
l+=1
```

Nota: este código no es eficiente, pero ayuda a entender el concepto de For Loop

20. Código para imprimir profundidad de 1000 a 1010 in incrementos de un metro

```
for number in range(1000, 1011, 1):  
    print(number)
```

21. Código para imprimir profundidad de 1000 a 1010 in incrementos de medio metro

```
for number in range(2000, 2011):  
    print(number / 2)
```

22. Código que detecte altos valores the ROP en una lista. Cuando un valor the ROP es detectado, se imprime una alerta indicando el valor the ROP y el indice en la lista. Genere los valores the ROP aleatoriamente.

```
import random  
  
rop_values = [random.randint(1, 100) for _ in range(50)]  
print(rop_values)  
print(len(rop_values))  
  
for index, rop_value in enumerate(rop_values):  
    if rop_value > 50:  
        print(f"High ROP: {rop_value}, at index {index}")
```

23. Ordenar elementos en una secuencia

```
a=[1,2,3,4,5]  
print(a)  
  
a.sort(reverse=True)  
print(a)  
  
a.sort(reverse=False)  
print(a)
```

24. Cortar elementos de una secuencia

```
a=[1,2,3,4,5,6,7,8,9]  
print(a)  
  
b=a[5:8]  
print(b)  
  
b=a[5:]  
print(b)  
  
b=a[:5]  
print(b)
```

```
b=a[:-2]
print(b)
```

```
b=a[-2:]
print(b)
```

25. Declarar un diccionario con parámetros de perforación

```
parametros={}

parametros={"depth":[1000,1001,1002,10003],"rop":[10,11,12,10]}
print(parametros)

print(parametros["depth"])
print(parametros["rop"])
```

26. Completar paso 16, explicando todo los componentes

```
import random

random.seed(42)

sample_data = [
    {
        'depth': 500 + i,
        'ROP': random.uniform(20, 40),
        'WOB': random.uniform(2000, 2500),
        'RPM': random.uniform(100, 120)
    }
    for i in range(10)
]

print("Sample Data:")
for row in sample_data:
    print(row)
```