

KATA 01

```
from datetime import date

print("Today's date is:" + str(date.today()))
```

✓ 0.1s Python

Today's date is: 2022-02-10

Construir un convertidor de unidades

El oficial superior> "Necesito una cosa más: un programa de conversión entre parsecs y años luz. Tener un programa de este tipo podría ser realmente útil en el puente para trazar nuestro curso".

Tú> "¡Lo haré!"

+ Code + Markdown

```
parsec = 11

lightyears = 3.26156 * parsec

print(str(parsec) + " parsec, is " + str(lightyears) + " lightyears")
```

✓ 0.6s Python

11 parsec, is 35.87715999999996 lightyears

KATA 02

Crear un entorno virtual

```
C:\Users\jesus>python -m venv env

C:\Users\jesus>env\scripts\activate

(env) C:\Users\jesus>
```

Instalar una biblioteca

```
(env) C:\Users\jesus>pip install python-dateutil
Collecting python-dateutil
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, python-dateutil
Successfully installed python-dateutil-2.8.2 six-1.16.0
WARNING: You are using pip version 21.2.4; however, version 22.0.3 is available.
You should consider upgrading via the 'C:\Users\jesus\env\Scripts\python.exe -m pip install --upgrade pip' command.

(env) C:\Users\jesus>pip freeze
python-dateutil==2.8.2
six==1.16.0
```

Desactivar un entorno virtual

```
(env) C:\Users\jesus>deactivate
C:\Users\jesus>
```

KATA 03

Para este ejercicio, escribirás una lógica condicional que imprima una advertencia si un asteroide se acerca a la Tierra demasiado rápido. La velocidad del asteroide varía dependiendo de lo cerca que esté del sol, y cualquier velocidad superior a 25 kilómetros por segundo (km/s) merece una advertencia.

Un asteroide se acerca, y viaja a una velocidad de 49 km/s.

```
# Añadir el código necesario para crear una variable que guarde la velocidad del asteroide.
# Escribe una expresión de prueba para calcular si necesita una advertencia.
# Agregue las instrucciones que se ejecutarán si la expresión de prueba es true o false.
asteroide = 49
if aseroide > 25:
    print("Advertencia")
else: ("Falsa advertencia")
```

[1] ✓ 0.2s Python

... Advertencia

Si un asteroide entra en la atmósfera de la Tierra a una velocidad mayor o igual a 20 km/s, a veces produce un rayo de luz que se puede ver desde la Tierra. Escribe la lógica condicional que usa declaraciones `if`, `else`, y `elif` para alertar a las personas de todo el mundo que deben buscar un asteroide en el cielo. ¡Hay uno que se dirige a la tierra ahora a una velocidad de 19 km/s!

```
# Agrega el código para crear una variable para un asteroide que viaja a 19 km/s
# Escribe varias expresiones de prueba para determinar si puedes ver el rayo de luz desde la tierra
# Agrega las instrucciones que se ejecutarán si las expresiones de prueba son True o False
asteroide = 19
if aseroide > 20:
    print("Luz en el cielo")
elif aseroide == 20:
    print("Luz en el cielo")
else:
    print("No hay nada en el cielo")
```

✓ 0.2s Python

No hay nada en el cielo

```
# Agrega el código para crear nuevas variables para la velocidad y el tamaño del asteroide
# Para probar el código, prueba con varias velocidades y tamaños
# Escribe varias expresiones de prueba o combinaciones de expresiones de prueba para determinar qué mensaje se debe envi
V_asteroide = 25
T_asteroide = 40
if V_asteroide > 25 and T_asteroide > 25:
    print("Advertencia")
elif T_asteroide >= 20:
    print("Luz en el cielo")
elif T_asteroide < 20:
    print("No hay luz en el cielo")
else: print("No hay luz en el cielo")
```

✓ 0.2s Python

Luz en el cielo

KATA 04

Gravity Facts about Ganymede

Planet Name: Mars

Gravity on Ganymede: 1.4300000000000002 m/s2

```
# Datos con los que vas a trabajar
name = "Mars"
gravity = 0.00143 # in kms
planet = "Ganymedes"

✓ 0.1s
```

Primero, crea un título para el texto. Debido a que este texto trata sobre la gravedad en la Tierra y la Luna, úsalo para crear un título significativo. Utiliza las variables en lugar de escribir.

```
# Creamos el título
title = f'datos de gravedad sobre {name}'

[13] ✓ 0.1s Python
```

Ahora crea una plantilla de cadena multilínea para contener el resto de la información. En lugar de usar kilómetros, debes convertir la distancia a metros multiplicando por 1,000.

```
# Creamos la plantilla
info = f"""{'-'*80}
name del planeta: {planet}
gravedad en {name}: {gravity * 1000} m/s2 """

[14] ✓ 0.1s Python
```

Finalmente, usa ambas variables para unir el título y los hechos.

```
# Unión de ambas cadenas
template = f"""{title.title()}
{info}
"""

print(template)
```

Finalmente, usa ambas variables para unir el título y los hechos.

```
# Unión de ambas cadenas
template = f"""{title.title()}
{info}
"""

print(template)
```

[15] ✓ 0.1s

```
-----
name del planeta: Ganymedes
gravedad en Mars: 1.4300000000000002 m/s2
```

Kata 10

```
PS C:\Users\jesus\Desktop\LaunchX> & C:/Users/jesus/AppData/Local/Programs/Python/Python310/python.exe c:/Users/jesus/Desktop/LaunchX/Open.py
Traceback (most recent call last):
  File "c:\Users\jesus\Desktop\LaunchX\Open.py", line 5, in <module>
    main()
  File "c:\Users\jesus\Desktop\LaunchX\Open.py", line 2, in main
    open("/path/to/mars.jpg")
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
PS C:\Users\jesus\Desktop\LaunchX>
```

```
PS C:\Users\jesus\Desktop\LaunchX> python Open.py
Traceback (most recent call last):
  File "C:\Users\jesus\Desktop\LaunchX\Open.py", line 5, in <module>
    main()
  File "C:\Users\jesus\Desktop\LaunchX\Open.py", line 2, in main
    open("/path/to/mars.jpg")
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
PS C:\Users\jesus\Desktop\LaunchX>
```

```
PS C:\Users\jesus\Desktop\LaunchX> python Config.py
Couldn't find the config.txt file!
PS C:\Users\jesus\Desktop\LaunchX>
```

```
PS C:\Users\jesus\Desktop\LaunchX> rm -f config.txt
Remove-Item : No se puede procesar el parámetro porque el nombre de parámetro 'f' es ambiguo. Las posibles coincidencias son: -Filter -Force.
En línea: 1 Carácter: 4
+ rm -f config.txt
+ ~~~
+ CategoryInfo          : InvalidArgument: (:) [Remove-Item], ParameterBindingException
+ FullyQualifiedErrorId : AmbiguousParameter,Microsoft.PowerShell.Commands.RemoveItemCommand
```

Interactive-1 X

Clear All Restart Interrupt ... Python 3.10.2 64-bit

```
def water_left(astronauts, water_left, days_left):
    daily_usage = astronauts * 11
    total_usage = daily_usage * days_left
    total_water_left = water_left - total_usage
    return f"Total water left after {days_left} days is: {total_water_left} liters"

[1] ✓ 0.2s
```

```
water_left(5, 100, 2)

[2] ✓ 0.1s
```

... 'Total water left after 2 days is: -10 liters'

```
water_left(5, 100, 2)
```

⊗ 4.3s

```
-----
RuntimeError                                Traceback (most recent call last)
Input In [2], in <module>
----> 1 water_left(5, 100, 2)

c:\Users\jesus\Desktop\LaunchX\Config.py in water_left(astronauts, water_left, days_left)
      4 total_water_left = water_left - total_usage
      5 if total_water_left < 0:
----> 6     raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
      7 return f"Total water left after {days_left} days is: {total_water_left} liters"

RuntimeError: There is not enough water for 5 astronauts after 2 days! ←
```

```
water_left("3", "200", None)
```

⊗ 0.3s

```
-----
TypeError                                Traceback (most recent call last)
Input In [3], in <module>
----> 1 water_left("3", "200", None)

c:\Users\jesus\Desktop\LaunchX\Config.py in water_left(astronauts, water_left, days_left)
      1 def water_left(astronauts, water_left, days_left):
      2     daily_usage = astronauts * 11
----> 3     total_usage = daily_usage * days_left
      4     total_water_left = water_left - total_usage
      5     if total_water_left < 0:

TypeError: can't multiply sequence by non-int of type 'NoneType' ←
```

During handling of the above exception, another exception occurred:

```
TypeError                                Traceback (most recent call last)
Input In [5], in <module>
----> 1 water_left("3", "200", None)

c:\Users\jesus\Desktop\LaunchX\Config.py in water_left(astronauts, water_left, days_left)
      5     argument / 10
      6 except TypeError:
      7     # TypeError will be raised only if it isn't the right type
      8     # Raise the same exception but with a better error message
----> 9     raise TypeError(f"All arguments must be of type int, but received: '{argument}'")
     10 daily_usage = astronauts * 11
     11 total_usage = daily_usage * days_left

TypeError: All arguments must be of type int, but received: '3' ←
```