

Hoja 1: Instrucciones de la shell de Linux, Literales y expresiones, Sangrado

Entregar en moodle el ejercicio 1 en un documento en formato pdf

Reflexionar sobre las posibles soluciones a los ejercicios 2 y 3 para debatir en clase.

1. Instrucciones de la shell de Linux (evaluable)

Objetivos

Escribir instrucciones de la shell de Linux para copiar datos.

Descripción

Dentro del directorio del usuario se dispone de dos directorios llamados instrucciones y experimento. El directorio instrucciones contiene, entre otros, ficheros de texto (con extensión ".txt"). El directorio experimento contiene a su vez, entre otros, varios ficheros con datos binarios, acabados en ".data", varios ficheros de hojas de datos acabados en ".csv" y un directorio llamado pictures, que contiene imágenes con extensión ".png".

Suponiendo que el directorio de trabajo inicial es el del usuario, y utilizando exclusivamente rutas relativas, escribir las órdenes linux/unix necesarias para crear un resumen de los datos almacenados, realizando los siguientes pasos:

- a) crear en el directorio del usuario un nuevo directorio llamado resumen
`mkdir resumen`
- b) copiar todos los ficheros acabados en ".txt" de instrucciones a resumen
`cp instrucciones/*.txt resumen/`
- c) mover todos los ficheros acabados en ".csv" de experimento a resumen
`mv experimento/*.csv resumen/`
- d) crear dentro de resumen un directorio llamado datos_binarios
`mkdir resumen/datos_binarios`
- e) copiar todos los ficheros acabados en ".data" de experimento a datos_binarios
`cp experimento/*.data resumen/datos_binarios/`
- f) mover el directorio pictures de experimento a resumen
`mv experimento/pictures resumen/`
- g) borrar todos los ficheros que estén todavía en experimento y sean del año pasado. Estos

Programación, Curso 2023-2024. Problemas

ficheros tienen en su nombre la secuencia "2023"
rm experimento/*2023*

2. Literales y expresiones

Objetivos

Familiarizarse con los literales y las expresiones de los tipos primitivos

Descripción

Escribir instrucciones (sueltas) Python que creen las variables x , y , z , r , s y t , y les den valores de acuerdo a las siguientes expresiones donde los números están descritos en castellano. No hacer simplificaciones. Utilizar notación científica (exponencial) cuando así se muestra.

$$x = \frac{33 \cdot 4,3 \cdot 10^3 + 2,5 \cdot 10^2 \cdot (27,6 - 37,8)}{12,7 + 10^{-2} \cdot 4,3}$$

$$y = (x - 23,4)^2$$

$$z = y \cdot 102.350.000 \quad (\text{y por } 102 \text{ millones trescientos cincuenta mil})$$

$$r = \left| \frac{x + y \cdot 10^{-3}}{100 - z} \right|$$

$$s = r + (2 + 3j)^2 \quad (\text{r más el número complejo } 2+3j \text{ elevado al cuadrado})$$

$$t = 10^{22} \cdot s^2 - 3 \cdot 10^4 j \quad (10^{22} \text{ por } s \text{ al cuadrado menos el número complejo } 0 + 3 \cdot 10^4 j)$$

Solución:

```
x:float = (33 * (4.3 * 10**3) + (2.5 * 10**2)) * (27.6 - 37.8) / (12.7 + 10**-2 * 4.3)
```

```
y:float = (x - 23.4) ** 2
```

```
z:float = y * 102350000
```

```
r:float = abs((x + y * 10**-3) / (100 - z))
```

```
s:complex = r + (2 + 3j) ** 2
```

```
t:complex = 10**22 * s**2 - 3 * 10**4j
```

3. Sangrado

Objetivos

Familiarizarse con el concepto de sangrado.

Descripción

Adaptar el siguiente módulo utilizando el sangrado que te parezca más adecuado. Tener en cuenta que

- El contenido de una función se sangra un nivel (4 espacios) a la derecha, respecto del encabezamiento.

Programación, Curso 2023-2024. Problemas

- En los comentarios de documentación, los argumentos se describen debajo de la palabra “Args:” o “Arguments:” o “Parameters:” y se ponen sangrados un nivel a la derecha.
- También en la documentación el valor retornado se pone debajo de la palabra “Returns:” y también se coloca sangrado un nivel a la derecha.
- Las líneas de continuación que tengan el salto de línea dentro de un paréntesis se sangran poniendo el texto de continuación justo un carácter a la derecha del paréntesis (de apertura.

```
# -*- coding: utf-8 -*-
"""
Simula el movimiento oscilatorio de una masa que
pende de un muelle, sin considerar rozamientos

Se usan unidades del sistema internacional:

@author: Michael
@date  : ene-2022
"""

from typing import Final

# Constantes:
# CTE_ELASTICA: Constante elástica del muelle, en N/m
# G: Gravedad terrestre, en m/s^2
CTE_ELASTICA: Final[float] = 0.52
G: Final[float] = 9.8

def fuerza_muelle(alt: float) -> float:
    """
    Calcula la fuerza que ejerce el muelle
    dada la altura de su extremo

    Args:
    alt: la altura del extremo del muelle, en m
    Returns:
    la fuerza que ejerce el muelle, en N
    """

    # Retornamos la fuerza calculada por medio de la ley de Hooke
    return -CTE_ELASTICA*alt

def avanza_tiempo(alt: float, vel: float, masa: float,
tiempo: float) -> tuple[float, float]:
    """
    Calcula la nueva altura y velocidad de la masa,
    transcurrido un tiempo especificado

    El tiempo debe ser pequeño, para poder considerar la
    aceleración constante en ese intervalo

    Args:
    alt: la altura actual del extremo del muelle, en m
```

Programación, Curso 2023-2024. Problemas

vel: la velocidad actual del extremo del muelle, m/s
masa: la masa del cuerpo que pende del muelle, en Kg
tiempo: el tiempo transcurrido, en s

Returns:

la nueva altura y velocidad de la masa,
después de transcurrido un tiempo t
"""

```
# Calculamos la fuerza aplicada sobre la masa, en N
fuerza = fuerza_muelle(alt) - masa*G
# Aceleración por la ley de newton
aceleracion = fuerza/masa
# Ecuacion del movimiento uniformemente acelerado
return (alt + vel*tiempo + (aceleracion*tiempo**2)/2,
        vel + aceleracion*tiempo)
```

```
def main():
    """
```

```
Simula el movimiento de una masa suspendida de un
muelle durante un tiempo, y pone en pantalla
altura y velocidad. Unidades del S.I.
    """
```

```
masa: float = 0.25
incremento: float = 0.01
alt: float = 0.2
vel: float = 0
```

```
# Avanzar la simulación y mostrar resultados
print("Altura=", alt, "m. Velocidad=", vel, "m/s")
alt, vel = avanza_tiempo(alt, vel, masa, incremento)
print("Altura=", alt, "m. Velocidad=", vel, "m/s")
alt, vel = avanza_tiempo(alt, vel, masa, incremento)
print("Altura=", alt, "m. Velocidad=", vel, "m/s")
```

Solución:

```
# -*- coding: utf-8 -*-
"""
```

```
Simula el movimiento oscilatorio de una masa que
pende de un muelle, sin considerar rozamientos
```

Se usan unidades del sistema internacional:

```
@author: Michael
@date : ene-2022
"""
```

```
from typing import Final
```

```
# Constantes:
# CTE_ELASTICA: Constante elástica del muelle, en N/m
```

Programación, Curso 2023-2024. Problemas

```
# G: Gravedad terrestre, en m/s^2
CTE_ELASTICA: Final[float] = 0.52
G: Final[float] = 9.8
```

```
def fuerza_muelle(alt: float) -> float:
```

```
    """
```

```
    Calcula la fuerza que ejerce el muelle
    dada la altura de su extremo
```

```
    Args:
```

```
    alt: la altura del extremo del muelle, en m
```

```
    Returns:
```

```
    la fuerza que ejerce el muelle, en N
```

```
    """
```

```
# Retornamos la fuerza calculada por medio de la ley de Hooke
```

```
return -CTE_ELASTICA*alt
```

```
def avanza_tiempo(alt: float, vel: float, masa: float, tiempo: float) -> tuple[float, float]:
```

```
    """
```

```
    Calcula la nueva altura y velocidad de la masa,
    transcurrido un tiempo especificado
```

```
    El tiempo debe ser pequeño, para poder considerar la
    aceleración constante en ese intervalo
```

```
    Args:
```

```
    alt: la altura actual del extremo del muelle, en m
```

```
    vel: la velocidad actual del extremo del muelle, m/s
```

```
    masa: la masa del cuerpo que pende del muelle, en Kg
```

```
    tiempo: el tiempo transcurrido, en s
```

```
    Returns:
```

```
    la nueva altura y velocidad de la masa,
    después de transcurrido un tiempo t
```

```
    """
```

```
# Calculamos la fuerza aplicada sobre la masa, en N
```

```
fuerza = fuerza_muelle(alt) - masa*G
```

```
# Aceleración por la ley de newton
```

```
aceleracion = fuerza/masa
```

```
# Ecuacion del movimiento uniformemente acelerado
```

```
return (alt + vel*tiempo + (aceleracion*tiempo**2)/2, vel + aceleracion*tiempo)
```

```
def main():
```

Programación, Curso 2023-2024. Problemas

```
pass
"""
Simula el movimiento de una masa suspendida de un
muelle durante un tiempo, y pone en pantalla
altura y velocidad. Unidades del S.I.
"""

masa: float = 0.25
incremento: float = 0.01
alt: float = 0.2
vel: float = 0

# Avanzar la simulación y mostrar resultados
print("Altura=", alt, "m. Velocidad=", vel, "m/s")
alt, vel = avanza_tiempo(alt, vel, masa, incremento)
print("Altura=", alt, "m. Velocidad=", vel, "m/s")
alt, vel = avanza_tiempo(alt, vel, masa, incremento)
print("Altura=", alt, "m. Velocidad=", vel, "m/s")
```