



programa "nota alfabética"

- pedir nota numérica
- si es menor de 5
- suspenso
- sino si es menor de 7
- aprobado
- sino si es menor de 9
- notable
- sino sobresaliente
- mostrar resultado

instrucción condicional múltiple

instrucción condicional múltiple

Sintaxis:

```
if expresión-booleana 1:  
    instrucción 1  
elif expresión-booleana 2:  
    instrucción 2  
elif expresión-booleana 3:  
    ...  
else:  
    instrucción n
```

Ejemplo:

```
if nota < 5:  
    calificacion = "suspenso"  
elif nota < 7:  
    calificación = "aprobado"  
elif nota < 9:  
    calificación = "notable"  
else:  
    calificación = "sobresaliente"
```

algunas funciones más

<code>len(cadena)</code>	devuelve la longitud de la cadena (número de caracteres)
<code>type(expr)</code>	devuelve el tipo de dato de la expresión (entero, real, booleano, carácter u otro)
<code>ord(carácter)</code>	devuelve el entero con el que se codifica y que se emplea para el orden de ordenación
<code>chr(entero)</code>	inversa de la anterior, dado un entero devuelve el carácter asociado
<code>range(n_final)</code> <code>range(n_inicial,n_final)</code> <code>range(n_inicial,n_final,n_incremento)</code>	generan secuencias de números enteros y se emplean sobre todo para bucles

```
# Muestra cuántos caracteres tiene una cadena
cadena=input("escribe una palabra o frase: ")
print("la cadena tiene",len(cadena),"caracteres")
```

```
# Ejemplos con la función type
```

```
a=5
```

```
print(type(a))
```

```
print(type(a>0))
```

```
print(type(a/2))
```

```
print(type(str(a)))
```

```
# Ejemplos con las funciones ord y chr
```

```
print(ord("A"),ord("B"),ord("C"))
```

```
print(ord("a"),ord("b"),ord("c"))
```

```
print(ord("á"),ord("ü"),ord("ñ"),ord("ß"))
```

```
print(ord("0"),ord("1"),ord("2"))
```

```
print(ord("."),ord(","),ord(";"))
```

```
print(ord("!"),ord("i"),ord(" "),ord("-"))
```

```
print(ord("α"),ord("ø"),ord("市"),ord("д"))
```

```
print(chr(65),chr(97),chr(48))
```

```
print(chr(241),chr(945),chr(26477))
```

```
print(chr(128512),chr(128518),chr(129315))
```

bucles for con la función range

```
for i in range(10):  
    print(i)
```

```
a=8  
for indice in range(a-3):  
    print(indice)
```

```
for j in range(4,10):  
    print(j)
```

```
for x in range(-3,6):  
    print(x)
```

```
for valor in range(1,9,2):  
    print(valor+1000)
```

```
for k in range(7,-2,-1):  
    print(k)
```

otras instrucciones para bucles: break y continue

"break" permite finalizar el bucle (salir de él) inmediatamente.
"continue" permite volver inmediatamente al inicio del bucle para realizar el siguiente ciclo del mismo.

Las dos pueden usarse tanto en los bucles "for" como en los bucles "while".

En general "continue" se usa menos.

break

El siguiente ejemplo ilustra el uso de la primera:

```
for i in range(5):  
    if i==3:  
        break  
    print(i)
```

La salida de este programa es 0, 1 y 2 uno en cada línea. Al ejecutarse la instrucción "break" el bucle se interrumpe y finaliza en vez de seguir con los valores 3 y 4.

continue

El siguiente ejemplo ilustra el uso de la segunda:

```
for i in range(5):  
    if i==3:  
        continue  
    print(i)
```

La salida de este programa es 0, 1, 2 y 4 uno en cada línea. Al ejecutarse la instrucción "continue" cuando "i" vale 3, no se ejecuta el resto del bucle en ese caso y se vuelve al principio para seguir con el siguiente valor de "i".

otras instrucciones para bucles: break y continue

```
# Pide números y calcula su suma, acaba con la entrada nula
suma=0
while True:
    dato=input("siguiente número: ")
    if dato=="":
        # no se ha introducido nada, salida del bucle
        break
    # dato válido, se añade a la suma
    suma=suma+float(dato)
# fin del bucle
print("la suma de todos es",suma)
```

otras instrucciones y funciones

<code>pass</code>	<code>instrucción nula, no hace nada</code>
<code>...</code>	<code>instrucción nula también</code>
<code>exit()</code>	<code>función para finalizar el programa</code>

ejemplo de uso de la función exit()

<otras instrucciones aquí>

```
# calculamos el cociente si se puede
```

```
if divisor==0:
```

```
    print("no se puede dividir por cero, fin del programa")
```

```
    exit()
```

```
# se puede calcular, prosigue el programa
```

```
cociente=valor/divisor
```

<más instrucciones aquí>

para hacer:

- programa que pida un número y diga si es cero, positivo o negativo
- programa que pida una nota numérica (por ejemplo 5.6) y muestre su alfabética (aprobado en el caso anterior)
- programa que pida una frase y muestre otra pero sin acentos
- programa que pida un número n y calcule la suma de los cubos del 1 al n
- programa que pida una temperatura en Celsius y la muestre en Fahrenheit
- programa que pida varias palabras o frases y diga cuántas se han introducido debiendo finalizar con la entrada vacía
- programa que pida números y muestre la suma final de todos debiendo finalizar con la entrada vacía
- programa que pida una frase y diga cuántas palabras tiene separadas por espacios

para hacer:

- programa que pida una frase y diga cuántas palabras tiene suponiendo que están separadas por cualquier carácter que no sea letra
- programa que pida palabras y muestre la más larga, finalizando con la entrada vacía
- programa que pida números positivos y muestre el mayor, finalizando con un número negativo
- programa que pida números y muestre el promedio de todos debiendo finalizar con la entrada vacía
- programa que pida una palabra y muestre si contiene alguna vocal
- programa que pida palabras y muestre cuántas contienen todas las vocales minúsculas sin acentuar, debiendo finalizar con la palabra vacía