ACTIVIDAD 1

SCRIPTS BÁSICOS

1. ¡Es una batalla de Pokémon! Tu tarea es calcular el daño que causaría un movimiento en particular usando la siguiente fórmula:

   daño = 50 * (ataque / defensa) * efectividad

   donde:

   - ataque = tu poder de ataque
   - defensa = la defensa del oponente
   - efectividad = la efectividad del ataque según el enfrentamiento (ver explicación a continuación)

   En cuanto a la efectividad, los ataques pueden ser súper efectivos, neutrales o poco efectivos según el enfrentamiento. Por ejemplo, el agua sería súper eficaz contra el fuego, pero no muy eficaz contra la hierba.

   - Súper efectivo: 2x daño
   - Neutral: 1x daño
   - No muy efectivo: 0,5x daño

   Para evitar que este desafío sea tedioso, solo habrá cuatro tipos: fuego, agua, hierba y electricidad. Aquí está la efectividad de cada enfrentamiento:

   - fuego > hierba
   - fuego < agua
   - fuego = electricidad
   - agua < hierba
   - agua < electricidad
   - hierba = electricidad

   La función que debes implementar y que se llamará calculaPupa recibirá:

   - Tu tipo
   - El tipo del oponente
   - Tu poder de ataque
   - La defensa del oponente

PISTA

Puedes codificar la efectividad de cada enfrentamiento en un objeto, de un modo similar a:

```
const efectividad = {
    fuego: {
      hierba: 2,
      agua: 0.5,
      electricidad: 1,
    }…
```

2. Realiza una función numeroDigitos que cuente el número de dígitos de un número sin usar strings. Por ejemplo:
   - `numeroDigitos(318) => 3`
   - `numeroDigitos(-314569) => 6`

RETO EXTRA. Busca información sobre la clase string y trata de hacer otra versión de la misma función de la manera más compacta posible.

3. Crea una función pasoAMonedas que pida un número real que representa una cantidad en euros y devuelva un array de 8 enteros representando el número mínimo de monedas de cada tipo necesario para pagar dicha cantidad, ordenadas por valor decreciente.
    - Posición 1: Número de monedas de 2 euros
    - Posición 2: Número de monedas de 1 euro
    - Posición 3: Número de monedas de 50 céntimos.
    - Posición 4: Número de monedas de 20 céntimos.
    - Posición 5: Número de monedas de 10 céntimos.
    - Posición 6: Número de monedas de 5 céntimos.
    - Posición 7: Número de monedas de 2 céntimos.
    - Posición 8: Número de monedas de 1 céntimo.

Ejemplo: ¿Qué cantidad desea convertir? → 5,26
5,26 euros son 2 x 2 euros, 1 x 1 euro, 1 x 20 céntimos, 1 x 5 céntimos, 1 x 2 céntimos.

2	1	0	1	0	1	2	0

En caso de que la cantidad introducida no sea un número real, se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

El script se repetirá hasta que el usuario introduzca la palabra "FIN" en lugar de un número real.

PISTAS
- Trabajar en céntimos te facilitará los cálculos
- El operador % te ayudará a saber cuánto te queda por cambiar
- Implementa y prueba en primer lugar la función `pasoAMonedas`. Una vez implementada y probada, implementa el cuerpo principal del script.

RETO EXTRA. Añadir una variable llamada **cajaMonedas** que contenga la cantidad de monedas de cada tipo de la cual disponemos (se inicializa al comenzar el script, preguntando al usuario):

Inicio: ¿Cuántas monedas de cada tipo tenemos inicialmente? → 5.

cajaMonedas

5	5	5	5	5	5	5	5

Primera pregunta: ¿Qué cantidad desea convertir? → 9,99
9,99 euros son 4 x 2 euros, 1 x 1 euro, 1 x 50 céntimos, 2 x 20 céntimos, 1 x 5 céntimos, 2 x 2 céntimos

cajaMonedas

1	4	3	3	5	4	3	5

Segunda pregunta: ¿Qué cantidad desea convertir? → 4,51
4,51 euros son 1 x 2 euros, 2 x 1 euro, 1 x 50 céntimos, 1 x 1 céntimo.
Se han agotado las monedas de 2 euros.

cajaMonedas

0	2	4	3	5	4	3	4

Tercera pregunta: ¿Qué cantidad desea convertir? → 9
No tenemos suficiente dinero para convertir dicha cantidad a monedas.