

## Ejercicios de repaso de Unity (Nivel básico)

Para todos estos ejercicios en los que no hay nada visual utiliza un game object vacío para colocar el script correspondiente.

### Sintaxis, variables y tipos de dato

En estos ejercicios no ha de haber en la escena más que un gameobject vacío al que le ponemos el script que hagamos como componente. Es decir no tienen representación visual, más allá de los mensajes que hagas aparecer en la consola con `print()` o `Debug.Log()`. Pero es importante que te fijas en todo lo que aparece en la consola, que leas los errores y te quedes con como solucionas cada tipo de error concreto. Así cuando aparezcan estos errores en programas más complejos te resultará más fácil saber como solucionarlos.

- Escribe un programa que defina cuatro variables de los siguientes tipos (una por cada tipo): int, float, string, bool. Ponles el nombre que quieras a cada una (asegurándote de que son nombres válidos) y dales los siguientes valores:
  - int -> -42
  - float -> 98.6
  - string -> "Hola mundo"
  - bool -> false

Haz que aparezcan estos valores en la consola. ¿Dónde tendrás que poner el código? ¿En la función Start? ¿En la función Update? ¿Por qué?

- Modifica el programa del ejercicio anterior de forma que después de que aparezcan en la consola los valores de las variables, estos cambien a los siguientes:
  - int -> 21
  - float -> 100.437
  - string -> "Hasta luego, mundo"
  - bool -> true

Haz que aparezcan estos valores en la consola una segunda vez.

- Escribe un programa que convierta una temperatura en grados centígrados en grados Fahrenheit, teniendo en cuenta que la fórmula de conversión es:  $F = C * 9/5 + 32$
- Escribe un programa que complementa al anterior y convierta grados Fahrenheit en grados centígrados.
- Escribe un programa que calcule el precio de la compra teniendo en cuenta que el kilo de manzanas es de 1,25€, el de plátano 0,7€, el de ciruelas 2,90€ y el de kiwi 2,45. Esta es la lista de la compra:
  - 2 Kg de manzanas
  - 1,5Kg de plátanos
  - 1/2 Kg de ciruelas
  - Cuarto y mitad de kiwis.

Utiliza variables tanto para los precios de cada producto como para sus cantidades.

- Escribe un programa que calcule la letra mensual de un coche usando la siguiente fórmula:

$$pago\ mensua = \left( interés + \frac{interés}{(1+interés)^{mesesPagados-1}} \right) \cdot precioTotal$$

El precio total es 12000€ y el interés 0.005. Calcula el pago del 1er mes del segundo año. (Busca en la página de Unity, en la ayuda de la clase `Mathf` como calcular una potencia).

### Condicionales

- Escribe un programa que calcule un número del 0 al 10 de forma aleatoria. Utilizar una serie de condicionales que:
  - comprueben si el número es menor que 3 o no y lo muestre en consola
  - comprueben si el número es menor que 3 o no y lo muestre en consola
  - comprueben si el número está entre 3 y 7 y lo muestre en consola
  - comprueben si el número está es menor que 3 o mayor que 7 y lo muestre en consola
- Escribe un programa que cree un campo numérico en el inspector y que calcule un número aleatorio. Si el número introducido en el inspector coincide con el calculado haz que muestre en consola el mensaje "Has acertado el número que estaba pensando" y si no que muestre "Has fallado".
- Como en el ejercicio anterior haz que aparezca un campo numérico en el inspector y que el programa muestre en la consola si el número es par o impar.
- Como en los ejercicios anteriores, haz que aparezca un campo numérico en el inspector. Si el número está entre 1 y 7 ha que en la consola se muestre el día de la semana correspondiente. Si no haz que muestre un mensaje diciendo que el número introducido no es válido.
- Haz que aparezca en el inspector un campo de texto. Si la cadena introducida tiene más de un carácter haz que el programa informe de que sólo debe tener un carácter. Si sólo tiene un carácter haz que muestre si es una vocal o una consonante.  
Aclaración: las cadenas de texto tienen, igual que los arrays una propiedad `Length`, que contiene el número de caracteres que hay en la cadena.
- Declara tres variables locales: `nombre`, `sexo` y `edad`. Construye una serie de condicionales que hagan que se muestre en la consola una frase del estilo "Verónica es una mujer adulta",

	de 0 a 16	de 16 a 30	a partir de 30
un hombre	un niño	un hombre joven	un hombre adulto
una mujer	una niña	una mujer joven	una mujer adulta

pero siguiendo la tabala siguiente, en función de los valores de dichas variables:

En el ejemplo de la frase, `nombre = "Verónica"`, `sexo = "mujer"` y `edad = 32`.

Comprueba todos los posibles casos cambiando el valor de las variables.

- Crea una escena en la que aparezcan un cubo y una esfera. Utilizando lo que sabes de componentes y la ayuda de Unity, obtén la información necesaria para averiguar si la esfera y el cubo están solapando y escribe un programa que, acoplado como componente a un gameobject vacío averigue y muestre en consola si éstos solapan.

### Bucles

- Escribe un programa que muestren en la consola los números del 10 al 20
- Escribe un programa que muestren en la consola los números pares del 7 al 57
- Escribe un programa que muestren en la consola los números pares del 50 al 10 en orden inverso
- Escribe un programa que muestren en la consola la suma de los números del 1 al 100
- Escribe un programa que muestren en la consola la suma de los números impares del 50 al 150

19. Escribe un programa que muestren en la consola los números múltiplos de 15 del 0 al 1000 (i, el contador, tiene que empezar en 0 y terminar en 1000. El bucle se tiene que repetir sólo 66 veces)

## Arrays

Elige el método de inicialización de arrays (elemento a elemento, con un bucle o con una expresión de inicialización) que más te convenga en cada caso:

20. Crea un array de `string` con las vocales y muéstralo en la consola.
  21. Tendrás que crear una función como la que ya tienes pero que reciba un array de `string`, en lugar de `int`
  22. Crea un array con los primeros 20 números pares y muéstralo en la consola.
  23. Crea un array con los números impares del 50 al 100 y muéstralo en la consola.
  24. Crea un array con 10 valores `true` y muéstralos en consola (tendrás que crear otra función que muestre en consola, esta vez, arrays de `bool`)
  25. Crea un array con 10 valores `true` y `false` alternos. Muéstralos en consola.
  26. Crea un array de 10 enteros que se rellene desde el inspector. Haz que se muestre en la consola pero en orden inverso (sin utilizar el método `Inverse()` de los arrays)
  27. Crea un array de 8 cadenas de texto que se llame `colores` y una variable `colorPreferido` que se puedan rellenar en el inspector. Haz que se muestre en la consola si el color preferido está o no en el array.
  28. Crea un array de 5 números que puedan tener decimales. Haz que se muestre en la consola la media de esos números.
  29. Crea un array de 10 enteros. Haz que se muestre en la consola el mayor de todos ellos.
- 
30. Crea un array con 10 instancias de un prefab con un cubo. Después añade los cubos que hay en el array a la escena, colocándolos en fila india y con una separación de 1 metro entre cubo y cubo. Fíjate en que digo que el array debe rellenarse con instancias del prefab. Esto significa que el array no se puede rellenar desde el inspector.
  31. Repite el ejercicio anterior pero rellenando el array de cubos desde el inspector, arrastrando el prefab a cada uno de los elementos. ¿Que diferencia hay entre los arrays de este ejercicio y el anterior?
  32. Repítelo de nuevo, pero ahora en vez de colocar todos los cubos de una vez, coloca uno cada segundo. Utiliza `Invoke()` para ello.
  33. Repite el ejercicio anterior pero, en vez de utilizar `Invoke()`, utiliza la función `Update()` y `Time.time`.
  34. Crea un array bidimensional para definir una sopa de letras de 10 por 10 letras. Utiliza dos bucles anidados para mostrar la sopa de letras en la consola (en filas y columnas, no elemento a elemento)
  35. Crea diferentes prefabs con cubos, esferas, cilindros o combinaciones de los mismos. Haz que no ocupen más que un metro cúbico de volumen (1x1x1)  
Construye un array bidimensional de 10x10 con instancias de esos prefabs y muéstralo en la escena colocado en un cuadrado sobre un plano de 20x20 metros distribuidos formando una rejilla.

## Listas (List<T>)

36. Crea una variable de tipo lista dinámica de colores con `List<string>`. Añade los colores "rojo", "verde" y "azul" al inicializar la lista. Después ve realizando las siguientes operaciones y mostrando en la consola una descripción y el resultado de la misma:
  - a) Añade el color "malva"
  - b) Añade los colores "anaranjado", "fucsia", "celeste" y "granate" usando una sola sentencia.
  - c) Limpia la lista para dejarla vacía.
  - d) Vuelve a añadir los colores "rojo", "verde" y "azul", usando una array.
  - e) Muestra en consola en que posición se encuentra el elemento "azul"
  - f) Vuelve a añadir el array de colores de antes y muestra en la consola en que posición se encuentra el elemento "verde". Explica de cual de ellos te está devolviendo la posición y porqué.
  - g) Añade de nuevo el color "verde"
  - h) Muestra en consola en que posición se encuentra la última instancia de "verde".
  - i) Inserta el color "amarillo" entre los primeros "azul" y "rojo".
  - j) Elimina el color "rojo".
37. Crea un programa que añada a la escena una esfera cada vez que pulses la tecla [Espacio]. Haz que las esferas generadas no solapen para que se pueda ver bien lo que está ocurriendo. Cuando hagas click con el ratón todas las esferas deben desaparecer de una en una en orden inverso al que aparecieron, a razón de una por segundo. ¿Por qué no puedes hacer este mismo programa con un array?
38. Haz como en el ejercicio anterior pero ahora para eliminar una esfera habrá que hacer click sobre ella. Para detectar sobre que esfera has hecho click puedes utilizar la siguiente función:

```
GameObject getSceneGameObject(string selectTag = null) {
    RaycastHit hit;
    GameObject pickedGO = null;
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    if ( Physics.Raycast (ray, out hit, 100.0f)) {
        GameObject selectedGO = hit.transform.gameObject;
        if (selectTag != null) {
            if (selectedGO.tag == selectTag) {
                pickedGO = selectedGO;
            }
        } else {
        }
    }
    return pickedGO;
}
```

pasándole como parámetro el tag que tienen los objetos que sean seleccionables (en este caso las esferas).

39. Crea un programa que genere un cubo que se mueva en la dirección Z ( $v = 2\text{m/s}$ ) en cada segundo. Haz que el programa compruebe cuantas unidades de Unity han recorrido todos los cubos que haya en la escena en cada fotograma. Haz que cada cubo pegue un salto cada 2 metros que avance.