

- **Tipos**:

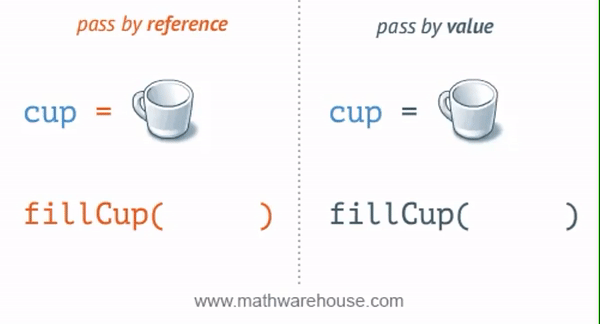
* Es la categoría general que clasifica las distintas formas en las que los datos pueden ser manejados en un lenguaje de programación. Esto incluye cómo se almacenan y manipulan los datos.

- **Por Valor**:

* Los tipos **por valor** almacenan directamente los datos en la memoria. Esto significa que cuando se copia un valor, se crea una copia independiente, y los cambios en una copia no afectan al original.
* Ejemplos comunes incluyen:
  + **Primitivos**: Tipos de datos básicos.
    - **Enteros**: Números enteros como 1, 2, -3.
    - **Reales**: Números con decimales como 3.14 o -2.5.
    - **Booleanos**: Valores lógicos como true o false.
  + **Estructuras**: Conjuntos de datos que pueden incluir diferentes tipos, definidos por el usuario.
  + **Enumerados**: Tipos definidos por el usuario que tienen un conjunto fijo de valores, como los días de la semana (Lunes, Martes, etc.).

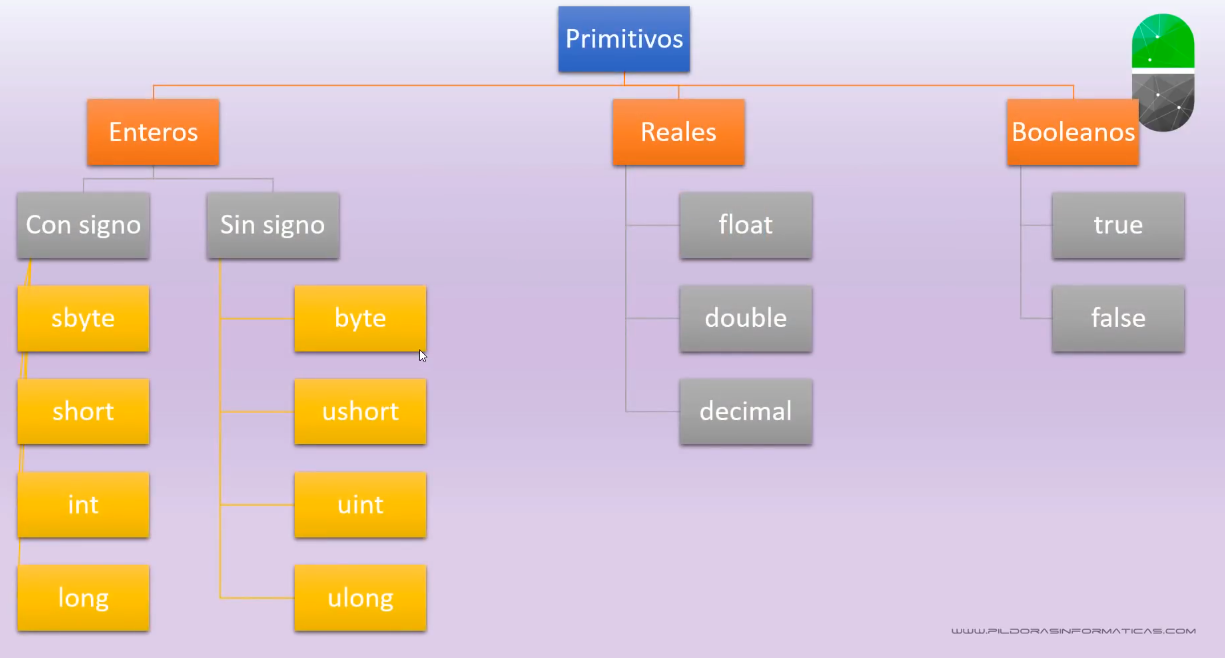
- **Por Referencia**:

* Los tipos **por referencia** no almacenan los datos directamente, sino que contienen una referencia (o dirección) a los datos reales en la memoria. Si varias referencias apuntan al mismo objeto, un cambio en una afecta a todas las demás.
* Ejemplo común: objetos y clases.



Cuando es por **valor**, la información de la variable se almacenan en una dirección de memoria diferente al recibirla en la funcion, por lo tanto si el valor de esa variable cambia no afecta la variable original, solo se modifica dentro del contexto de la función.

Cuando es por **referencia**, la variable que se recibe como parámetro en la función apunta exactamente a la misma dirección de memoria que la variable original por lo que si dentro de la función se modifica su valor también se modifica la variable original.



**1. Enteros**

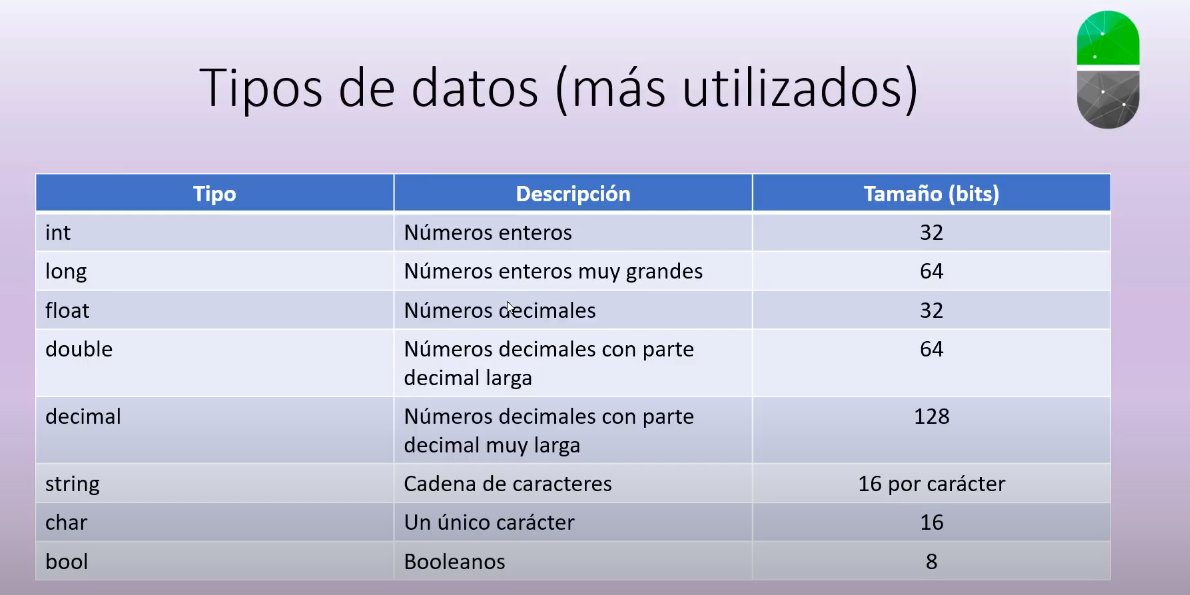
* Representan números enteros, y se subdividen en dos categorías:
  + **Con signo**:
    - Pueden almacenar valores positivos y negativos.
    - Tipos:
      * sbyte: Entero de 8 bits con rango de -128 a 127.
      * short: Entero de 16 bits con rango de -32,768 a 32,767.
      * int: Entero de 32 bits con rango de -2,147,483,648 a 2,147,483,647.
      * long: Entero de 64 bits con rango de -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807.
  + **Sin signo**:
    - Solo pueden almacenar valores positivos.
    - Tipos:
      * byte: Entero de 8 bits con rango de 0 a 255.
      * ushort: Entero de 16 bits con rango de 0 a 65,535.
      * uint: Entero de 32 bits con rango de 0 a 4,294,967,295.
      * ulong: Entero de 64 bits con rango de 0 a 18,446,744,073,709,551,615.

**2. Reales**

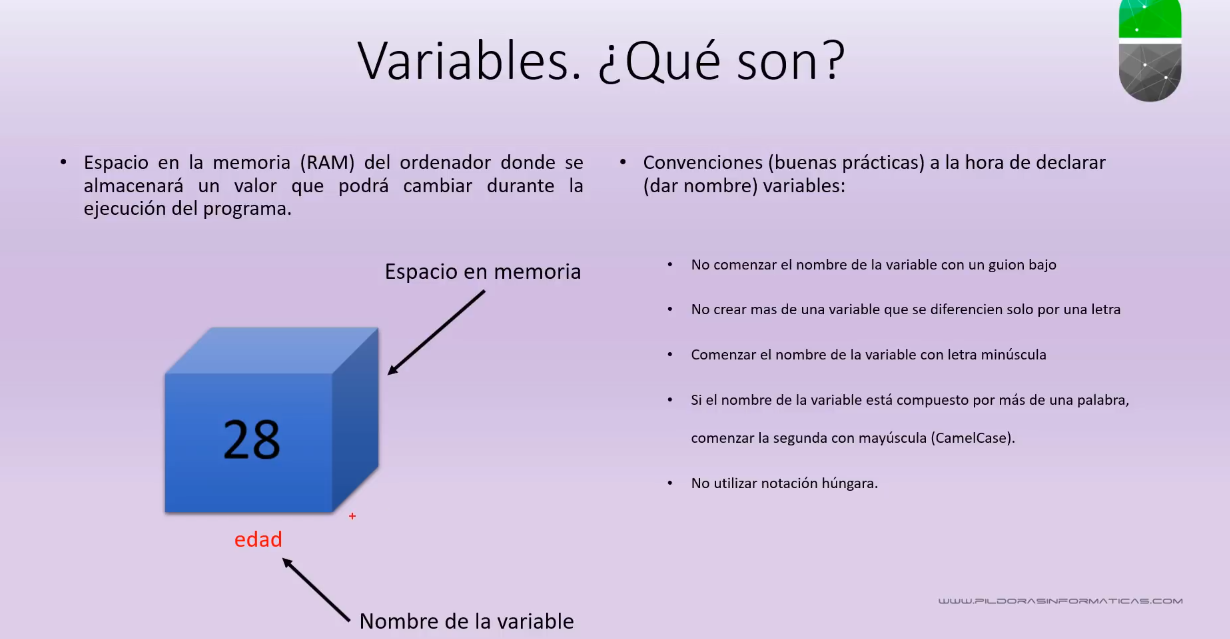
* Representan números decimales (con punto flotante).
* Tipos:
  + float: Precisión simple (32 bits), ideal para cálculos menos precisos.
  + double: Precisión doble (64 bits), más exacto para valores decimales.
  + decimal: Alta precisión (128 bits), usado principalmente en cálculos financieros.

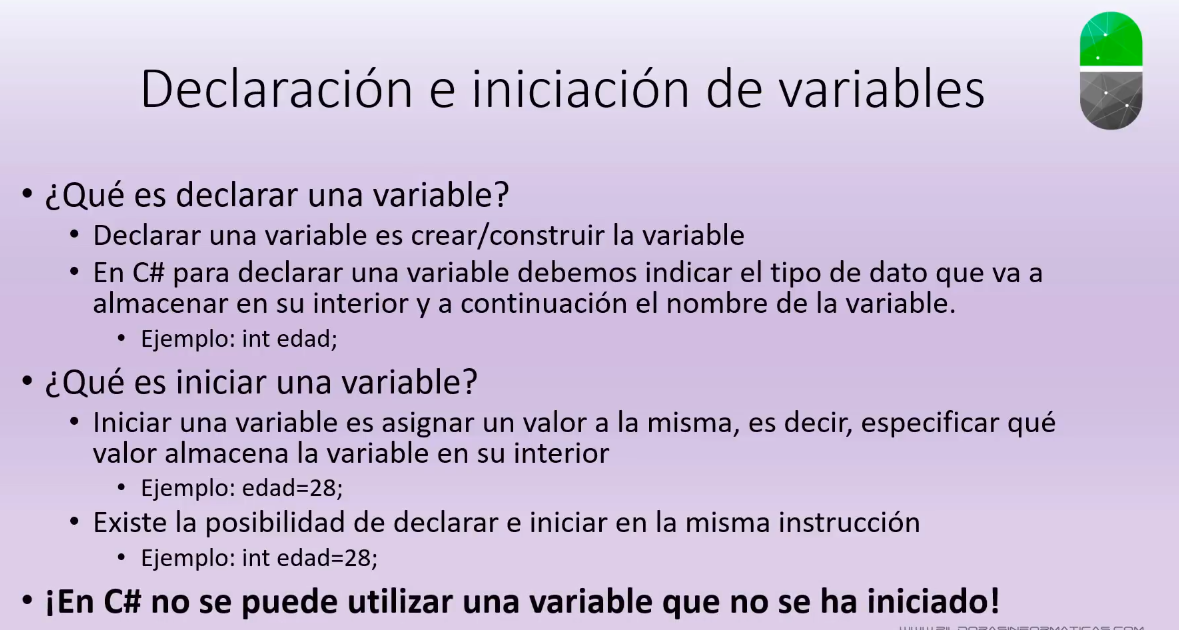
**3. Booleanos**

* Representan valores lógicos, solo tienen dos posibles valores:
  + true: Verdadero.
  + false: Falso.



* String es un tipo de dato por referencia.
* Char se considera una categoría mas en los tipos de datos.
* El tamaño indica cuanta RAM voy a ocupar al declarar una variable de determinado tipo.





* Inicializar = definir