

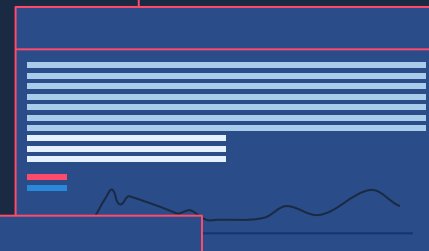


Tecnicatura en Programación Universitaria

Comisión

B

Clase 02

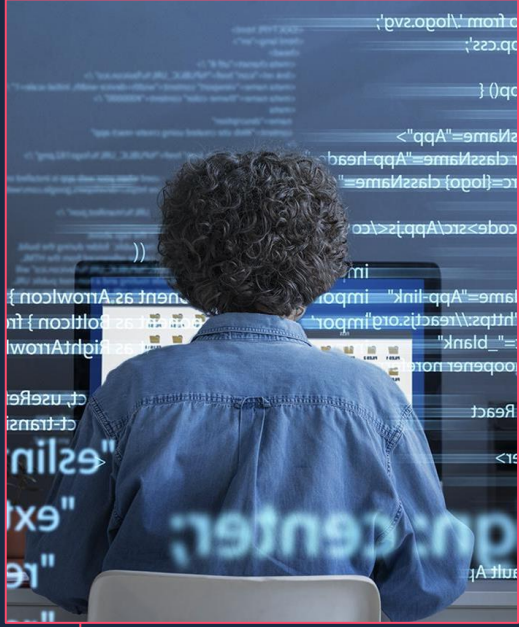
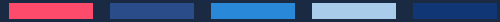


Clase Practica

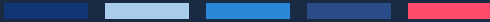
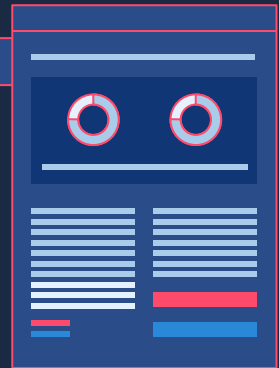
Martes 16 de Abril 2024

Ing. Oviedo Codigoni Carlos Nicolas





...Variables...



Variables en C

ENTEROS

- **int:** Entero con signo.
- **unsigned int:** Entero sin signo.
- **short:** Entero corto con signo.
- **unsigned short:** Entero corto sin signo.
- **long:** Entero largo con signo.
- **unsigned long:** Entero largo sin signo.
- **long long:** Entero largo largo con signo.
- **unsigned long long:** Entero largo largo sin signo.

REALES

- **float:** Número de punto flotante.
- **double:** Número de punto flotante de doble precisión.
- **long double:** Número de punto flotante de precisión extendida.

CARACTERES

- **char:** Un solo carácter.
- **signed char:** Carácter con signo.
- **unsigned char:** Carácter sin signo.

Espacio en la Memoria - sizeof()

sizeof es una función muy útil en C para obtener información sobre el tamaño en bytes de tipos de datos y variables, lo que es especialmente útil para la gestión de memoria y la portabilidad del código entre diferentes sistemas.

ENTEROS

32 bits

- **int:** 4 bytes
- **unsigned int:** 4 bytes
- **short:** 2 bytes
- **unsigned short:** 2 bytes
- **long:** 4 bytes
- **unsigned long:** 4 bytes
- **long long:** 8 bytes
- **unsigned long long:** 8 bytes

64 bits

- **int:** 4 bytes
- **unsigned int:** 4 bytes
- **short:** 2 bytes
- **unsigned short:** 2 bytes
- **long:** 8 bytes
- **unsigned long:** 8 bytes
- **long long:** 8 bytes
- **unsigned long long:** 8 bytes

REALES

- **float:** 4 bytes
- **double:** 8 bytes
- **long double:** 12 bytes (*puede variar*)

- **float:** 4 bytes
- **double:** 8 bytes
- **long double:** 16 bytes (*puede variar*)

CARACTERES

- **char:** 1 byte
- **signed char:** 1 byte
- **unsigned char:** 1 byte

- **char:** 1 byte
- **signed char:** 1 byte
- **unsigned char:** 1 byte

```
# include<exercise.h>
```

Usando la función `sizeof()` y `printf ()` de la librería `stdio.h` realizar una tabla donde en la primera columna muestre el tipo de variable y en la segunda columna muestre el espacio de memoria que se almacena en la computadora.

!!! Recuerda !!!

\n : Realiza un salto de línea.

\t : Realiza un espacio (tabular) a la izquierda.

```

1 #include <stdio.h>
2
3 int main() {
4     printf("-----\n");
5     printf("Tipo de variable\n");// Enteros (Integer Types)
6     printf("-----\n");
7     printf("\n");
8     printf("-----\n");
9     printf("Enteros\t\tEspacio (bytes)\n");
10    printf("-----\n");
11    printf("int:\t\t\t%lu\n", sizeof(int));
12    printf("unsigned int:\t\t\t%lu\n", sizeof(unsigned int));
13    printf("short:\t\t\t%lu\n", sizeof(short));
14    printf("unsigned short:\t\t\t%lu\n", sizeof(unsigned short));
15    printf("long:\t\t\t%lu\n", sizeof(long));
16    printf("unsigned long:\t\t\t%lu\n", sizeof(unsigned long));
17    printf("long long:\t\t\t%lu\n", sizeof(long long));
18    printf("unsigned long long:\t\t\t%lu\n", sizeof(unsigned long long));
19
20    // Reales (Floating-Point Types)
21    printf("-----\n");
22    printf("Reales\t\tEspacio (bytes)\n");
23    printf("-----\n");
24    printf("float:\t\t\t%lu\n", sizeof(float));
25    printf("double:\t\t\t%lu\n", sizeof(double));
26    printf("long double:\t\t\t%lu\n", sizeof(long double));
27
28    // Caracteres (Character Types)
29    printf("-----\n");
30    printf("Caracteres\t\tEspacio (bytes)\n");
31    printf("-----\n");
32    printf("char:\t\t\t%lu\n", sizeof(char));
33    printf("signed char:\t\t\t%lu\n", sizeof(signed char));
34    printf("unsigned char:\t\t\t%lu\n", sizeof(unsigned char));
35
36    return 0;
37 }

```

Función Scanf ()

La función scanf en C se utiliza para leer datos de entrada desde la consola o desde otro flujo de entrada, como un archivo, y almacenarlos en variables

```
scanf(formato, &variable1, &variable2, ...);
```

- **formato:** Es una cadena de caracteres que especifica cómo se deben interpretar los datos de entrada. Los **especificadores de formato** comienzan con el símbolo de porcentaje (%) seguido de un carácter que indica el tipo de datos que se espera leer.
- **&variable1, &variable2, etc.:** Son las direcciones de memoria de las variables donde se almacenarán los datos leídos.

Especificadores de Formato

- **%d:** Lee un valor entero decimal (int).
- **%f:** Lee un valor flotante (float).
- **%lf:** Lee un valor flotante de doble precisión (double).
- **%c:** Lee un único carácter (char).
- **%s:** Lee una cadena de caracteres (string).
- **%u:** Lee un valor entero sin signo (unsigned int).
- **%ld:** Lee un valor entero largo (long).
- **%lu:** Lee un valor entero largo sin signo (unsigned long).
- **%lld:** Lee un valor entero largo largo (long long).
- **%llu:** Lee un valor entero largo largo sin signo (unsigned long long).
- **%x, %X:** Lee un valor hexadecimal (int).
- **%o:** Lee un valor octal (int).
- **%e, %E:** Lee un número en notación científica (float o double).
- **%g, %G:** Lee un número en notación estándar o en notación científica, dependiendo de la

Ejemplos de uso de Scanf ()

ENTEROS

```
1 #include <stdio.h>
2
3 int main() {
4     // Declaración de la variable para almacenar un carácter
5     char inicial;
6
7     // Solicitar al usuario que ingrese una inicial
8     printf("Ingrese una inicial: ");
9
10    // Leer la inicial ingresada por el usuario y almacenarla en la variable
11    // Espacio antes de %c para ignorar espacios en blanco
12    scanf(" %c", &inicial);
13
14    // Mostrar la inicial ingresada por el usuario
15    printf("La inicial ingresada es: %c\n", inicial);
16
17    return 0;
18 }
```

REALES

```
1 #include <stdio.h>
2
3 int main() {
4     // Declaración de la variable para almacenar un entero
5     int numero;
6
7     // Solicitar al usuario que ingrese un número entero
8     printf("Ingrese un número entero: ");
9
10    // Leer el número entero ingresado por el usuario y almacenarlo en la variable
11    scanf("%d", &numero);
12
13    // Mostrar el número ingresado por el usuario
14    printf("El número ingresado es: %d\n", numero);
15
16    return 0;
17 }
```

CARACTERES

```
1 #include <stdio.h>
2
3 int main() {
4     // Declaración de la variable para almacenar un carácter
5     char inicial;
6
7     // Solicitar al usuario que ingrese una inicial
8     printf("Ingrese una inicial: ");
9
10    // Leer la inicial ingresada por el usuario y almacenarla en la variable
11    scanf(" %c", &inicial);
12
13    // Mostrar la inicial ingresada por el usuario
14    printf("La inicial ingresada es: %c\n", inicial);
15
16    return 0;
17 }
```

include<exercise.h>

Usando las funciones vista hasta el momento realizar los siguientes ejercicios ...

1. **Imprimir la edad:** Solicita al usuario su edad y luego la imprime.
2. **Imprimir un carácter ASCII:** Solicita al usuario un número entero y luego imprime el carácter ASCII correspondiente.
3. **Imprimir un entero en hexadecimal:** Solicita al usuario un número entero y luego lo imprime en formato hexadecimal.
4. **Imprimir un número en notación científica:** Solicita al usuario un número en notación científica y luego lo imprime.
5. **Suma de dos números:** Solicita al usuario dos números enteros y muestra la suma.
6. **Multiplicación de dos números:** Solicita al usuario dos números enteros y muestra el resultado de su multiplicación.
7. **Calculadora de área de un rectángulo:** Solicita al usuario la base y la altura de un rectángulo y muestra su área.
8. **Calculadora de área de un triángulo:** Solicita al usuario la base y la altura de un triángulo y muestra su área.
9. **Conversión de Celsius a Fahrenheit:** Solicita al usuario una temperatura en Celsius y muestra su equivalente en Fahrenheit.
10. **Área y circunferencia de un círculo:** Solicita al usuario el radio de un círculo y muestra su área y circunferencia.