

Elementos que se introducen en esta Unidad:

char, String

Desde la aparición del ordenador se plantearon lenguajes de programación para sus acciones. A partir de ese momento, hubo mucha investigación en la entrada y la salida de texto. De esta forma se expandió el campo de la programación a lo que fueron los primeros intentos por IA (Inteligencia Artificial). Una serie de programas en los que uno escribía lo que deseaba y el mismo programa, a duras penas, respondía algo mas o menos acorde. Esta clase de programas ha evolucionado, más allá de que existen todavía problemas con esta clase de ideas.

Más adelante, se experimentó con programas (preferentemente en la web) de traducción de idiomas, con resultados extraordinarios por parte de algunos servidores, que aún existen en la actualidad.

Esta unidad no discute temas acerca de traducción de idiomas o inteligencia artificial, sin embargo el texto es el tipo de dato mas común creado y modificado por el programa. El texto creado por e-mails, publicaciones y sitios web es un vasto recurso de información que puede ser almacenada y presentada en forma de datos.

-Caracteres

Los datos del tipo char pueden almacenar tipografías como A, e, 5 y %. El nombre char es el diminutivo para *character* cuyo significado es carácter, y este tipo de datos se distingue por se escritos entre comillas (' '). Las variables tipo char son declaradas como variables tipo int o tipo float.

```
char a = 'n';      //Asigna 'n' a la variable a
char b = n;        //ERROR - Sin comillas, n es una variable
char c = "n";      //ERROR - Las comillas dobles (" ") definen a n como un
                  //String, no un char
char d = 'not';    //ERROR - Las variables tipo char pueden almacenar
                  //solo un carácter
```

El siguiente ejemplo crea una nueva variable tipo char, asigna valores y los imprime en la consola:

```
char letra = 'A'; //Declara una variable letra y asigna el valor de 'A'
println(letra);   //Imprime "A" en la consola
letra = 'B';      //Asigna 'B' a la variable letra
println(letra);   //Imprime "B" en la consola
```

Muchos valores char tienen un valor número que les corresponde por el código de la tabla ASCII. Por ejemplo, A es 65, B es 66, C es 67, y así. Puede probar cada código usando la función `println()`:

```
char letra = 'A';
println(letra); //Imprime "A" en la consola
int n = letra;  //Asigna el valor numérico de A a la variable n
println(n);     //Imprime 65 en la consola
```

Es posible articular funciones matemáticas con valores alfabéticos. En el siguiente ejemplo se utiliza un ciclo FOR para imprimir el alfabeto de la A a la Z, por aumentar los valores char con la función de incremento:

```
char letra= 'A';      //Declara una variable tipo char y asigna el valor de A
for (int i = 0; i < 26; i++) {
    print(letra);      //Imprime un carácter en la consola
    letra++;           //Suma 1 al valor del carácter
}
println('.');          //Agrega un punto al final del alfabeto
```

-Palabras y Oraciones

Al utilizar una variable de tipo `String` pueden almacenarse palabras y oraciones completas. Para distinguir los datos `String` del resto del programa se utilizan comillas dobles (`""`). Las dobles comillas diferencian "s" como `String` de 's' como carácter. Los datos tipos `String` son diferentes a los del tipo `int`, `float` o `char`, ya que se trata de un *objeto*, o sea un tipo de dato que puede almacenar muchos elementos y funciones. Las variables tipo `PImage` y `PFont` también son objetos. Las variables tipo `String` se declaran y se asignan como cualquiera:

```
String a = "Anónimo";    //Asigna "Anónimo" a la variable a
String b = 'E';          //ERROR - Las comillas ' ' deben ser dobles.
String c = "E";          //Asigna "E" a c
string d = "E";          //ERROR - String debe comenzar con mayúsculas
```

Los siguientes ejemplos muestran algunos usos básicos de las variables tipo `string`:

```
//Los datos tipo String pueden contener largos textos
String s1 = "Rakete bee bee?";
String s2 = "Rrrrrrrrrrrrrrrrrummmmpffff tillffff tooooo?";
println(s1);          //Imprime "Rakete bee bee?"
println(s2);          //Imprime "Rrrrrrrrrrrrrrrrrummmmpffff tillffff tooooo?"

//Datos tipo String combinados con el operador +
String s3 = "Rakete ";
String s4 = "rinnzekete";
String s5 = s3 + s4;
println(s5);          //Imprime "Rakete rinnzekete"
```