

14. Langage : La classe String : une classe pour gérer facilement les chaînes de caractères !

Une fois que l'on sait utiliser une variable contenant un caractère, on a logiquement envie de pouvoir manipuler des chaînes de caractères.

Utiliser un tableau de variables **char** ?

La solution qui vient à l'esprit est de vouloir créer un tableau de variables de type char, ce qui est effectivement une solution possible, mais qui présente un gros inconvénient : la taille du tableau ne pourra pas varier au cours du programme ! Dès lors impossible d'ajouter simplement des caractères à moins de créer d'emblée un grand tableau...

Une bien meilleure solution : utiliser la classe String !

La lourdeur de la gestion des chaînes de caractères à l'aide d'un tableau a poussé l'équipe du langage Arduino à créer une classe spéciale dédiée à la gestion des chaînes de caractères : c'est un des domaines où toute la puissance du langage Arduino est évidente !

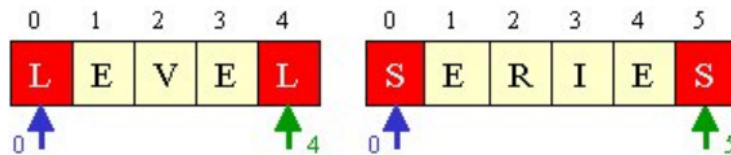
La classe String : un conteneur de chaîne de caractères !

La classe String c'est tout d'abord un « **conteneur** » de **chaîne de caractères** de **taille variable** : pas besoin d'en préciser la taille initiale et on pourra à tout moment lui ajouter un caractère supplémentaire.

La classe String : de nombreuses fonction très pratiques !

Mais ce n'est pas tout : la classe String va mettre à notre disposition toute une **série de fonctions très très pratiques** pour utiliser les chaînes de caractères : ajouter une chaîne, connaître la taille de la chaîne, se positionner à un endroit précis de la chaîne, éliminer les espaces, extraire une sous-chaîne, etc...

En un mot, **souplesse maximale** !



Déclaration d'un objet de la classe String

Un objet de la classe String se déclare tout simplement de la même façon qu'une variable sous la forme :

```
String chaineReception=""; // déclare un objet String vide
```

Usage avancé : noter que l'on ne parle plus ici de variable, mais d'objet car on crée une instance (l'objet créé) de la classe String (le moule). Le concept d'objet est plus large que celui de variable et associe un espace mémoire ainsi que des fonctions spécifiques. En pratique, on peut considérer qu'un objet String est une sorte de « super-variable ».

Opérations de base sur d'un objet String

On peut ajouter très simplement une chaîne de caractère ou des Strings entre eux :

```
maChaine=maChaine+ «test »; // ajoute une chaîne au String
```

```
maChaine=maChaine+ maChaine2; // ajoute une chaîne au String
```

Quelques fonctions de la classe String

- **charAt()** : renvoie le caractère à la position voulue
- **compareTo()** : compare le String à une chaîne
- **endsWith()** : teste si le String se termine par une sous-chaîne
- **indexOf()** : recherche la position d'une chaîne
- **length()** : renvoie la longueur de la chaîne
- **setCharAt()** : modifie le caractère à la position donnée
- **startsWith()** : teste si le String commence par une sous-chaîne
- **substring()** : extrait une sous-chaîne à la position donnée
- **trim()** : élimine les espaces

Objet String et les fonctions **print()** ou **println()**

Un objet String s'utilise très simplement avec les fonctions **print()** et **println()** :

```
Serial.print(maChaine); // affiche le String
```

15. Pour info : toutes les façons valides d'initialiser un String

```
String stringOne = "Hello String";           // en utilisant une chaîne de caractères
String stringOne = String('a');              // conversion d'un caractère simple en objet String appelé stringOne
String stringTwo = String("This is a string"); // conversion d'une chaîne de caractère en objet String appelé
stringTwo
String stringOne = String(stringTwo + " with more"); // concaténation d'un objet String et d'une chaîne
String stringOne = String(13);                // conversion d'un nombre en base 10 par défaut
String stringOne = String(analogRead(0), DEC); // conversion d'une valeur int en base 10
String stringOne = String(45, HEX);           // conversion de la valeur 45 en base hexadécimale
String stringOne = String(255, BIN);          // conversion de la valeur 255 en base binaire
String stringOne = String(millis(), DEC);     // conversion d'une valeur long en base 10
```

Pas de panique !

Vous n'avez pas besoin de retenir toutes les fonctions de la classe String par cœur : nous les utiliserons au fur et à mesure des besoins !
Retenez simplement que c'est un outil très puissant pour manipuler des chaînes de caractères !