

# Ce qu'il faut savoir sur la commande TryParse

La commande **TryParse** permet de convertir une chaîne de caractères en un type autre que string. L'intérêt par rapport à un **Parse** classique est que l'instruction TryParse va d'abord essayer de convertir et **retourner le succès (ou l'échec) de la conversion dans un booléen**. Par ce biais, **le code ne plante pas en cas de conversion ratée, et il laisse le choix au développeur de la manière gérer la situation**.

```
string myStringValue = "100";
int number;

bool success = int.TryParse(myStringValue, out number);
```

Le bout de code suivant va essayer de convertir la valeur "100" enregistrée dans la variable myStringValue. La **fonction TryParse retournera un booléen qui indiquera si la conversion a réussi** (true) ou non (false). Si la conversion a réussi, la valeur sera enregistrée dans la variable passée en second paramètre à savoir dans l'exemple ci-dessus la variable **number**.

Il est également possible de convertir une chaîne de caractères en un autre type que **integer** en utilisant :

- Pour les booléens : bool.TryParse
- Pour les float: float.TryParse
- Etc

Le fonctionnement est identique.

## Je ne comprends pas le mot clef out ?

De manière très simplifiée (et donc forcément un peu fausse), le mot clef out permet d'enregistrer la valeur parsée dans une variable qui a été déclarée préalablement si la conversion peut s'effectuer.

De cette manière, **avec une seule méthode** (TryParse), nous pouvons à voir à la fois **le booléen indiquant la réussite de la conversion et la variable convertie**. Il n'est pas possible de retourner plus d'un élément dans une fonction. Il aurait fallu sinon encapsuler les deux informations (la réussite et la valeur convertie) dans un objet qui aurait été retourné par la méthode tryparse.

Dans les prochaines vidéos, nous allons expliquer le fonctionnement d'un programme et cela nous permettra de comprendre ce qu'il se passe exactement avec le mot clef **out** et pourquoi il est utilisé.