

Les utilisateurs du système d'exploitation Windows se sont peut-être déjà familiarisés avec l'organisation hiérarchique des données dans le Registry, qui structure logiquement des valeurs à l'aide d'objets.

De manière similaire, les objets configurés par TypoScript sont organisés en une structure arborescente. L'outil de gabarit **TypoScript Object Browser** (le module **Web** → **Gabarit**) représente cette hiérarchie via une interface utilisateur graphique.

5.2.2 TSref

Nous voudrions à présent vous rappeler brièvement ce qu'est la référence TSref de TypoScript ; il s'agit d'une sorte de bible pour ceux qui travaillent quotidiennement avec TypoScript, puisque vous y trouvez des descriptions précises de tous les objets, propriétés et fonctions disponibles. Ayez toujours la référence à portée de main lorsque vous travaillez avec TypoScript ! TSref est disponible en ligne sur TYPO3.org et est disponible en format OpenOffice ou PDF. Si vous ne possédez pas encore TSref, le moment est venu de le télécharger !

Référence 342678

5.2.3 Digression : TypoScript et PHP

Afin d'acquérir une meilleure compréhension des aspects techniques, la section suivante traite de la relation entre TypoScript et PHP. Passez directement à la section 5.2.4 si vous le préférez.

Même si, en théorie, vous pouvez créer une sortie sous n'importe quelle forme en utilisant PHP, le *TypoScript Frontend Engine* (TSFE) propre à TYPO3 est utilisé par défaut lorsqu'un site Web est appelé via le fichier `index.php` (`tslib/index_ts.php`). Il analyse l'information dans l'enregistrement de gabarit de l'arborescence des pages du site Web².

PHP traite ici les objets et les valeurs des enregistrements de gabarit qui sont structurés par l'intermédiaire de TypoScript.

- L'information est placée par le système dans un tableau PHP multidimensionnel à l'aide de `t3lib_TSparger` (`t3lib/class.t3lib_tsparger.php`). Ce tableau est disponible dans certaines applications et fonctions de TYPO3.
- Si de l'information qui n'est pas utilisée par les fonctions des classes TYPO3 est placée dans le tableau PHP, elle se comporte comme une variable inutilisée déclarée en PHP : elle est ignorée et n'induit pas d'erreurs à la sortie.

Un exemple permet de clarifier cette notion : dans le code TypoScript abstrait suivant, l'information analysée par PHP est placée dans un tableau multidimensionnel.

```
monObjet.propriete1 = valeur_x
monObjet.propriete2 = valeur_y
monObjet.propriete2.propriete3 = valeur_z
```

En PHP, le tableau serait directement créé, comme suit :

```
$TS['monObjet.']['propriete1'] = 'valeur_x';
$TS['monObjet.']['propriete2'] = 'valeur_y';
$TS['monObjet.']['propriete2.']['propriete3'] = 'valeur_z';
```

²Le processus détaillé de restitution du frontend est décrit aux sections 5.7 et 7.5

ou alternativement :

```
$TS = array(
    'monObjet.' => array(
        'propriete1' => 'valeur_x',
        'propriete2' => 'valeur_y',
        'propriete2.' => array (
            'propriete3' => 'valeur_z'
        )
    )
)
```

La fonction `debug()` propre à TYPO3 peut aussi afficher un tableau avec le résultat suivant :

Figure 5.3:
Affichage par la
fonction TYPO3
`debug()` de la syntaxe
Typo3

monObjet.	propriete1	valeur_x
	propriete2	valeur_y
	propriete2.	propriete3

TYPO3 fournit un outil pour afficher et éditer TypoScript, appelé *TypoScript Object Browser*. Il affiche le code d'exemple de la manière suivante :

Figure 5.4:
Affichage du code TS
par le TypoScript
Object Browser

```
SETUP ROOT
├─ [config]
├─ [includeLibs]
├─ [plugin]
├─ [tt_news]=< plugin.tt_news
├─ [tt_address]=COA
├─ [tt_content]
├─ [page]=PAGE
├─ [monObjet]
│   └─ [propriete1]=valeur_x
│       └─ [propriete2]=valeur_y
│           └─ [propriete3]=valeur_z
├─ [resources]=
├─ [sitetitle]=
└─ [types]
```

Exemple : HRULER

Référence 762761

Les effets combinés de TypoScript et de PHP sont illustrés par l'objet de contenu (cObject) HRULER qui trace une ligne horizontale. Considérons le gabarit TypoScript suivant :

```
page = PAGE
page.typeNum = 0
page.20 = HRULER
page.20 {
    lineThickness = 10
    lineColor = #e6e6e6
    spaceLeft = 100
    spaceRight = 100
}
```