

# CheatSheet – Mes règles de ternaires

## 1. Ternaires pour initialiser une valeur

Forme générale :

```
$valeur = ($condition) ? valeur_si_vrai : valeur_si_faux;
```

Exemple simple :

```
$color = ($user->isAdmin()) ? 'red' : 'blue';
```

## 2. Ternaires pour construire un objet depuis une string

Forme générale :

```
$string = ($condition) ? 'valeur personnalisée' : 'valeur par défaut';  
$obj = new Objet($string);
```

Exemple pour DateTimeImmutable :

```
$date_str = ($annee && $mois) ? "$annee-$mois-01" : "";  
$date = new DateTimeImmutable($date_str);
```

✅ Fonctionne parce que `new DateTimeImmutable('')` = date du moment.

## 3. Ternaires sur retour de fonction ou paramètres

Quand tu veux passer un paramètre optionnel dans une fonction :

```
fonction(($option !== null) ? $option : 'valeur par défaut');
```

→ Alternative plus moderne (PHP 7+) : opérateur de fusion null ??

```
fonction($option ?? 'valeur par défaut');
```

## 📌 4. Ternaires dans un return

Forme rapide pour des retours conditionnels :

```
return ($isActive) ? 'Actif' : 'Inactif';
```

## 📌 5. Ne jamais imbriquer plusieurs ternaires dans un seul

Exemple à éviter :

```
$status = ($user->isAdmin()) ? 'admin' : ($user->isModerator() ? 'modo' : 'user');
```

⚠️ Illisible → Préférer un if/elseif/else classique si plusieurs cas.

## 📦 En résumé – Ma règle de base :

Question	Réponse
1 condition binaire simple ?	→ OK pour ternaire.
Plusieurs cas ?	→ Utiliser if/else pour la lisibilité.
Création d'objet dépendant d'une string ?	→ Ternaire string + new Objet(\$string).
Fusion simple de valeurs par défaut ?	→ Utiliser ?? (null coalescing).

## 🔥 Bonus : Mon petit "mantra" dev perso

"Si je dois réfléchir plus de 2 secondes pour lire mon ternaire → c'est trop, je dois faire un if/else."