



## Synthetic

```
}  
@var text_quoted = "Hello !";  
@var text = Hello !;  
@var list = [1,2, "Salut", Le monde !, [true, 0], 4];  
@var dict = {  
  cle : valeur,  
  autre_cle : {  
    // ...Autre imbrication  
  }  
}
```

## Le flux de sortie

Le flux de sortie est traité par la fonction `@out`, elle ne prend qu'un seul argument. Elle affiche son unique paramètre en même temps qu'elle le traite comme du code du langage Synthetic. Dans le cas d'utilisation d'une variable, il faut précéder son nom avec le caractère "\$".

```
@out($bonjour);
```

À l'intérieur de la fonction `@out`, on peut exécuter du code Synthetic, en réalité, la fonction `@out` n'est qu'un espace d'exécution en ligne de Synthetic.

```
@out(  
  @var bonjour = "Bonjour tout le monde !";  
  @out($bonjour); //Affiche "Bonjour tout le monde !"  
);
```

## Les structures conditionnelles

Comme dans tout langage de programmation, les structures conditionnelles permettent de tester des valeurs pour exécuter des codes différents. Synthetic reconnaît `@if`, `@elif` et `@else`.

```
@if($version == 1){  
  @out(La version est basique);  
}@elif($version < 3){  
  @out(La version plutot cool !)  
}@else{  
  @out(La version est trop cool !)}
```