

LES VARIABLES

Utilisation des variables

- Pour utiliser les variables et afficher le contenu associé, il faut faire précéder le nom de la variable par un \$.

```
#!/bin/bash
PRENOM="Jordan"
NOM="Thomas"
echo "Bonjour $PRENOM $NOM et bienvenu"
```

```
./script.sh
```

```
Bonjour Jordan Thomas et bienvenu
```

Utilisation des variables (2)

- Lorsque vous souhaitez inclure une variable dans un mot par exemple, vous pouvez utiliser les { }

```
#!/bin/bash
PRENOM="Jordan"
NOM="Thomas"
AGE="16"
echo "Bonjour $PRENOM ${NOM}, vous avez
${AGE}ans"
```

```
./script.sh
```

```
Bonjour Jordan Thomas, vous avez
16ans
```

Assigner les sorties de commandes à une variable

- Il est tout à fait possible d'assigner la sortie standard d'une commande à une variable en la mettant entre parenthèses.

```
#!/bin/bash
PRENOM="Jordan"
NOM="Thomas "
MACHINE=${hostname}
echo "Bonjour $PRENOM $NOM et bienvenu
sur la machine ${MACHINE}."
```

```
./script.sh
```

Assigner les sorties de commandes à une variable (2)

- Une autre syntaxe est également possible dans d'autres scripts qui utilisent le ` (alt + 7) à la place du \$().

```
#!/bin/bash
PRENOM="Jordan"
NOM="Thomas "
MACHINE=`hostname`
echo "Bonjour $PRENOM $NOM et bienvenu
sur la machine ${MACHINE}."
```

```
./script.sh
```

Attention au nom des variables !

- Attention à la syntaxe que l'on utilise pour les noms des variables.
- Quelques règles à respecter :
 - Les variables ne peuvent pas commencer par un chiffre mais peuvent en contenir.
 - Exemple : **1BONNEVARIABLE** n'est pas un bon nom
 - Exemple : **UNEBONNE7VARIABLE** est un nom correct
 - Les variables ne peuvent pas contenir de tiret (-)
 - Exemple : **UNE-BONNE-VARIABLE** n'est pas un bon nom
 - Exemple : **UNE_BONNE_VARIABLE** est un nom correct
 - Les variables ne peuvent contenir que des underscores, majuscules, minuscules et chiffres :
 - Exemple : **UNE_BONNE@VARIABLE** n'est pas un bon nom
 - Exemple : **Une_bonne_7_variable** est un nom correct