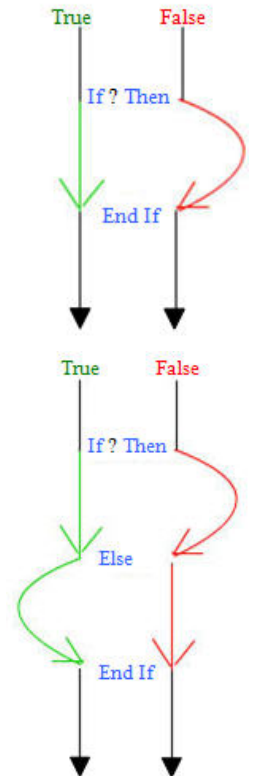


Mémento Bonus 1 VBA infodauphine.com

Branchements, cas et conditions

Itinéraire

- Avec If: VBA exécute le bloc d'instructions du **Then** si la condition est **vraie**. Dans tous les cas, l'exécution continue **à partir du End If ensuite**.
- Avec If **Else**: VBA exécute le bloc d'instructions **du Then** si la condition est **vraie** ou **du Else** si elle est **fausse**. Dans les 2 cas, **continue à partir du End If ensuite**.
- **End If**: là où les chemins se rejoignent: je met après le End If ce que je veux forcément faire.
- On peut **imbriquer**, c'est-à-dire mettre un If ou If/Else dans le bloc du Then ou dans le bloc du Else. Cela permet de faire **des sous-cas** et de placer des instructions qui seront faites **dans les 2 sous-cas** (par exemple entre les End If)
- Avec des ElseIf, uniquement le premier cas vrai est exécuté. Je me sers des **conditions précédentes** comme de filtres, car elles sont **forcément fausses**.



Réflexion sur les branchements

- Vérifier qu'on n'oublie pas de cas, et faire attention au cas où il ne doit **rien se passer**
- Se demander si les cas sont **incompatibles**, ou si plusieurs peuvent être vrais en même temps (on les met alors dans des **If séparés**). S'ils sont **incompatibles**, utiliser des **ElseIf** et se souvenir que les cas précédents sont **faux**.
- S'ils sont incompatibles, se demander si ils sont **exhaustifs**, ou si il y a des cas où il ne se passe rien qu'on laisse de côté (et à ce moment là, pas de Else)
- Toujours se demander dans quel **ordre** on a intérêt à faire ses branchements, et sous quel **structure**, pour éviter de dupliquer du code et raccourcir les conditions.
- Toujours **se méfier du Else**. Le traître du semestre qui conduit à beaucoup d'erreurs.

Opérateur booléens

- **And** et **Or** sont les opérateurs d'**addition booléenne**. Pour **Or** l'addition est True dès qu'une des booléenne est True. Pour **And**, il faut que les 2 soient True.
- on peut en mettre autant qu'on veut, mais uniquement **dans des conditions**, et on utilise les **parenthèses** pour les priorités
- **Not** inverse une booléenne: True devient False et inversement.
- **différent**: <> **inférieur ou égal**: <= **supérieur ou égal**: >=

Boucle et multiples

- Avec un **branchement dans une boucle**, je **teste à chaque tour**. Ce qui est hors du branchement sera fait à chaque tour, dans le branchement uniquement quand la condition est vraie
- If $x \text{ Mod } n = 0$ Then : si x est **divisible** par n

www.infodauphine.com

Exécute sur le site tous les exemples, et ton propre code, grâce à **EasyVBA**