

Les structures conditionnelles Dart

Table des matières

I. Structure conditionnelle simple : if	3
II. Exercice : Quiz	4
III. Structure conditionnelle alternative : if... else	5
IV. Exercice : Quiz	5
V. Structure conditionnelle généralisée : if... else if	6
VI. Exercice : Quiz	7
VII. Opérateurs ternaires	8
VIII. Exercice : Quiz	8
IX. Essentiel	9
X. Auto-évaluation	9
A. Exercice	9
B. Test	10
Solutions des exercices	12

I. Structure conditionnelle simple : if

Environnement de travail : Dartpad

Pré-requis : pas de pré-requis

Durée du cours : 1 h

Contexte

Avec des « *si* », on mettrait Paris en bouteille !

Cette expression bien connue signifie qu'avec des hypothèses, tout est possible mais bien inutile. Et pourtant, le conditionnel, en programmation, est **indispensable** !

Le conditionnel permet de changer le comportement d'une fonction si une condition est remplie, et se trouve dans (presque) tous les programmes.

Les structures conditionnelles en Dart peuvent être de plusieurs natures (simple, conditionnelle alternative ou conditionnelle généralisée). Nous allons détailler chacune d'elle dans ce cours.

L'utilité du « *si* » ne sera bientôt plus un secret pour vous !

Définition

Une **structure conditionnelle** est simple lorsque l'exécution du programme ne dépend que d'une seule condition.

La condition if permet d'exécuter une série d'instructions uniquement si une condition est remplie (c'est-à-dire que la condition retourne vrai).

Exemple

```
1 if (10 > 5) {  
2     print('La condition est vraie');  
3 }
```

10 > 5 retournera toujours vrai, et if (10 > 5) mènera toujours à la réalisation des instructions liées à la structure conditionnelle, et donc à l'affichage de « *La condition est vraie* ».

Si, par exemple, nous avions mis 2 > 5 en condition, celle-ci serait fausse et l'expression dans la condition if ne serait jamais exécutée.

Remarque

Il est tout à fait possible de mettre des structures conditionnelles les unes dans les autres !

Exemple

```
1 if (i < 2) {  
2     if (i == 1) {  
3         print("i est inférieure à 1 !");  
4     }  
5 }
```

Dans cet exemple, si *i* est inférieure à 2, la première condition est vraie et les instructions de la première structure sont exécutées. Ici, ces instructions sont à nouveau d'une structure conditionnelle, et si *i* est égale à 1, la condition est remplie et le code à l'intérieur de cette seconde structure peut être exécuté (à savoir l'affichage de « *i est inférieure à 1* »).

Exercice : Quiz

[solution n°1 p.13]

Question 1

Qu'est-ce qu'une structure conditionnelle simple ?

- ☐ Une structure conditionnelle simple est une structure dont la condition est toujours vraie
- ☐ Une structure conditionnelle est simple lorsque l'exécution du programme ne dépend que d'une seule condition
- ☐ Une structure conditionnelle simple mène à la fin de l'exécution du programme

Question 2

Si je souhaite qu'une série d'instructions au sein d'une structure conditionnelle soit exécutée, ma condition doit retourner :

- ☐ « Vrai »
- ☐ « Faux »

Question 3

Puis-je « imbriquer » des structures conditionnelles ?

- ☐ Oui, autant que je veux !
- ☐ Seulement deux fois

Question 4

Que se passe-t-il si la condition d'une structure conditionnelle simple n'est pas remplie ?

- ☐ L'exécution du programme s'arrête. Il faut que la condition soit remplie
- ☐ Le programme passe à la suite des instructions, après la structure conditionnelle

Question 5

```
1 if (i < 1) {
2     if (I == 1) {
3         print(I est inférieur à 1 !')
4     }
5 }
```

Qu'est-ce que mon programme va afficher ?

- ☐ Rien du tout !
- ☐ i est inférieure à 1

III. Structure conditionnelle alternative : if... else

Dans la structure conditionnelle simple, si la condition n'est pas remplie, nous passons simplement à la suite des instructions. Et si nous pouvions proposer une alternative, qui s'exécuterait uniquement si la première condition est fausse ?

Dans ce cas, nous utiliserions else.

Exemple

```
1 if (5 < 10) {  
2   print('La condition est vraie');  
3 }  
4 else {  
5   print('La condition est fausse');  
6 }
```

Dans cet exemple, l'instruction dans else ne s'exécute que si la première condition contenue dans le if n'est pas remplie.

Exercice : Quiz

[solution n°2 p.14]

Question 1

La structure conditionnelle alternative permet :

- ☐ De continuer l'exécution du programme
- ☐ De proposer une alternative si la condition if n'est pas remplie, et qui ne s'exécutera que dans ce cas de figure
- ☐ De proposer une alternative qui s'exécutera dans tous les cas

Question 2

Si la condition if n'est pas remplie, la condition else s'exécutera forcément.

- ☐ Vrai
- ☐ Faux

Question 3

```
1 var i = 5;  
2 if (i = 4) {  
3     print("Youpi");  
4 } else {  
5     print("Youpla !");  
6 }
```

Qu'est-ce que mon programme va afficher ?

- ☐ Youpi
- ☐ Youpla
- ☐ Youpi youpla

Question 4

Puis-je proposer plusieurs alternatives avec une structure conditionnelle simple ?

- ☐ Oui
- ☐ Non

V. Structure conditionnelle généralisée : if... else if

Fondamental

La structure conditionnelle généralisée permet de faire plusieurs évaluations de condition à la suite et exécute uniquement l'expression dont la condition est vraie.

Exemple

```
1 double note = 3,5;
2 string avis;
3 if (note >= 4,5) {
4     avis = 'Excellent';
5 } else if (note >= 3,5) {
6     avis = 'Très bon';
7 } else if (note >= 2,5) {
8     avis = 'Bon';
9 } else if (note >= 1,5) {
10    avis = 'Moyen';
11 } else
12 {
13     avis = 'Bof';
14 }
15 print(avis);
```

Dans cet exemple, la seconde condition est vraie, et c'est donc les instructions de celle-ci qui s'exécutent. Les conditions suivantes seront ignorées même si leur condition était également remplie.

Fondamental

Pour éviter l'utilisation de if et else if de nombreuses fois, dans le cas où de multiples conditions doivent être vérifiées, il est possible d'utiliser l'instruction switch et d'ainsi avoir une syntaxe plus légère.

Exemple

```
1 switch(expression){
2     case a:
3         print(a);
4         break;
5     case b:
6         print(b);
7         break;
8     case c:
9         print(c);
10        break;
11    default:
12        //le code par default
13 }
```

L'instruction break permet de sortir de l'instruction switch, et de ne pas évaluer le reste des cas (comme si nous utilisions else if). Il est à noter que les cas évalués ne peuvent être que des constantes, et il n'est pas possible d'évaluer une condition telle que $i < 10$ (« case $i < 10$: » n'est pas valide).

Exercice : Quiz

[solution n°3 p.15]

Question 1

Que permet une structure conditionnelle généralisée ?

- ☐ D'appliquer une condition à l'ensemble d'un programme
- ☐ D'appliquer une condition à l'ensemble des variables d'un programme
- ☐ De créer plusieurs évaluations de conditions dans une structure conditionnelle

Question 2

Si une condition d'une structure conditionnelle est vraie :

- ☐ Les conditions suivantes ne seront pas évaluées
- ☐ Toutes les conditions suivantes seront évaluées
- ☐ Seulement la condition else de fin sera également évaluée

Question 3

```
1 Var livre = 'Harry Potter';  
2 if (livre == 'Le Seigneur des Anneaux')  
3     { print('L'auteur est J. R. R. Tolkien');  
4 } else if (livre == 'Dune') {  
5     print('L'auteur est Franck Herbert');  
6 } else  
7 {  
8     print('L'auteur est J. K. Rowling');  
9 }
```

Qu'est-ce que le programme affichera ?

- ☐ L'auteur est J. R. R. Tolkien
- ☐ L'auteur est Franck Herbert
- ☐ L'auteur est J. K. Rowling

Question 4

Est-il possible d'utiliser l'instruction switch pour obtenir le même résultat qu'avec le programme de la question précédente ?

- ☐ Oui
- ☐ Non
- ☐ Ça dépend

Question 5

Si toutes les conditions alternatives sont remplies, que se passe-t-il ?

- ☐ Seules les instructions de la première condition sont exécutées
- ☐ Toutes les instructions de toutes les conditions sont exécutées
- ☐ Aucune instruction n'est exécutée

VII. Opérateurs ternaires

Fondamental

L'opérateur ternaire est considéré comme un raccourci pour une structure conditionnelle. Il permet de créer cette structure en une seule ligne de code !

Exemple

```
1 var résultat = 5 < 10 ? « La condition est vraie » : « La condition est fausse »;
```

La condition est ici `5 < 10`, qui est donc toujours vraie. C'est donc la première expression qui sera exécutée : la variable prend pour valeur « *La condition est vraie* ».

Fondamental

Il existe également une seconde syntaxe encore plus simple si nous souhaitons uniquement vérifier si une expression est différente de null avant l'exécution d'une autre instruction.

Exemple

```
1 var i = null;
2 var a;
3 a = i ?? 'Seconde expression';
```

Puisque `i` est ici égale à null, la variable `a` ne prendra pas la valeur de `i` mais la valeur « *Seconde expression* ».

Exercice : Quiz

[solution n°4 p.16]

Question 1

Un opérateur ternaire est :

- ☐ Une structure conditionnelle simplifiée
- ☐ Une structure conditionnelle permettant des actions supplémentaires

Question 2

Si je souhaite vérifier qu'une expression n'est pas nulle avant d'en utiliser la valeur, quelle syntaxe puis-je utiliser ?

- ☐ `expr1 ? expr1 : expr2`
- ☐ `expr1 ?? expr2`
- ☐ Aucune des deux

Question 3

```
1 var i = 3 < 4 ? 4 : 5;
```

Quelle valeur prendra `i` ?

- ☐ 4
- ☐ 5
- ☐ 3

Question 4


```
1 var a = null;  
2 var b = a ?? 'Seconde expression'
```

Quelle valeur prendra b ?

- ☐ null
- ☐ 'Seconde expression'
- ☐ 'a'
- ☐ a

Question 5

Je souhaite imbriquer plusieurs structures conditionnelles - qu'est-ce qui est possible ?

- ☐ `expr1 ? (Expr2 ?? expr3) : expr4`
- ☐ `if (expr1) { if(expr2) {} }`
- ☐ Aucune des deux propositions

IX. Essentiel

LA STRUCTURE CONDITIONNELLE SIMPLE

- Une structure conditionnelle est simple lorsque l'exécution du programme ne dépend que d'une seule condition.
- Il est possible d'imbriquer des structures conditionnelles de tout type.

LA STRUCTURE CONDITIONNELLE ALTERNATIVE

La structure conditionnelle alternative permet de proposer une alternative au cas où la condition n'est pas remplie.

LA STRUCTURE CONDITIONNELLE GÉNÉRALISÉE

- La structure conditionnelle généralisée permet de proposer plusieurs alternatives.
- Dès qu'une des alternatives est vraie, les conditions suivantes ne sont pas évaluées.

LES OPÉRATEURS TERNAIRES

- Un opérateur ternaire est une structure conditionnelle simplifiée.
- Il est surtout utilisé dans l'attribution de valeurs à des variables.
- Il existe deux syntaxes pour les opérateurs ternaires :
 - `Expr1 ? Expr2 : Expr3` : si Expr1 est faux, c'est Expr2 qui s'exprime, sinon Expr3.
 - `Expr1 ?? Expr2` : si Expr1 est null, alors c'est Expr2 qui s'exprime, sinon Expr1.

X. Auto-évaluation

A. Exercice

Grâce aux conditions, il nous est possible de faire répondre nos programmes à une multitude de questions à notre place. Par exemple, pour un très grand nombre, il peut nous être difficile de déterminer très rapidement s'il s'agit d'un multiple de 3. Avec la puissance de calcul d'un ordinateur, en revanche, c'est presque instantané !

Question

[solution n°5 p.17]

Créez un programme nous indiquant si la valeur d'une variable est un multiple de 2 ou un multiple de 3, ou si elle n'est divisible ni par l'un ni par l'autre.

Si la variable est égale à 3 ou à 2, nous souhaiterions également afficher « *La variable est égale à 3* » ou « *La variable est égale à 2* ».

Exemple : si nous mettons var nombre = 3, le programme affiche « *La variable est un multiple de 3 !* » puis « *La variable est égale à 3 !* ».

Si nous mettons var nombre = 4, le programme affiche « *La variable est un multiple de 2 !* » puis « *La variable n'est pas 2 !* ».

Si nous mettons var nombre = 7, le programme affiche « *La variable n'est ni un multiple de 3 ni un multiple de 2* ».

B. Test

Exercice 1 : Quiz

[solution n°6 p.18]

Question 1

Qu'est-ce qui s'affiche lorsque le programme suivant est exécuté ?

```
1 main () {
2
3     var i = 7;
4
5     if (i < 8) {
6         print(« Première condition »);
7     }
8     else if (i == 7) {
9         print(« Seconde condition »);
10    }
11    else {
12        print(« Ne rentre dans aucune condition »);
13    }
14 }
```

- ☐ Première condition
- ☐ Seconde condition
- ☐ Ne rentre dans aucune condition

Question 2

Je souhaite créer un programme affichant « *Premier cas* » si la variable x est égale à 4, mais aussi « *Deuxième cas* » si la variable est paire. Quelles instructions puis-je utiliser ?

- ☐ if... else
- ☐ if... else if
- ☐ if... if

Question 3

Quel est le bon opérateur pour une condition parmi les propositions suivantes ?

- ☐ ==
- ☐ =
- ☐ !=

Question 4

Quelle est la bonne version du programme ci-dessous avec l'utilisation de l'instruction switch ?

```
1 var x = 10;
2 if (x == 5) {
3     print(« x est égale à 5 »);
4 }
5 else if (x == 8) {
6     print(« x est égale à 8' »);
7 }
8 else if (x == 10) {
9     print(« x est égale à 10 »);
10 }
11 else {
12     print(« x ne rentre dans aucune des conditions »);
13 }
```

```
1 Switch (x) {
2     case 5:
3         print(« x est égale à 5 »);
4         break;
5     case 8:
6         print(« x est égale à 8' »);
7         break;
8     case 10:
9         print(« x est égale à 10 »);
10        break;
11    default:
12        print(« x ne rentre dans aucune des conditions »);
13 }
```

```
1 Switch (x) {
2     case 5:
3         print(« x est égale à 5 »);
4     case 8:
5         print(« x est égale à 8' »);
6     case 10:
7         print(« x est égale à 10 »);
8     default:
9         print(« x ne rentre dans aucune des conditions »);
10 }
```

- ☐ La première
- ☐ La deuxième
- ☐ La troisième
- ☐ Les deux premières
- ☐ Les trois

Question 5


Puis-je imbriquer des structures conditionnelles dans une structure conditionnelle généralisée ?

- ☐ Oui
- ☐ Non
- ☐ Pas toujours

Solutions des exercices


Exercice p. 4 Solution n°1**Question 1**

Qu'est-ce qu'une structure conditionnelle simple ?

- ☐ Une structure conditionnelle simple est une structure dont la condition est toujours vraie
- ☒ Une structure conditionnelle est simple lorsque l'exécution du programme ne dépend que d'une seule condition
- ☐ Une structure conditionnelle simple mène à la fin de l'exécution du programme
-  Une structure conditionnelle simple ne possède qu'une seule condition et ne propose pas d'alternatives.


Question 2

Si je souhaite qu'une série d'instructions au sein d'une structure conditionnelle soit exécutée, ma condition doit retourner :

- ☒ « Vrai »
- ☐ « Faux »
-  Les instructions liées à une structure conditionnelle ne seront exécutées que si la condition est vraie.


Question 3

Puis-je « imbriquer » des structures conditionnelles ?

- ☒ Oui, autant que je veux !
- ☐ Seulement deux fois
-  Nous pouvons imbriquer autant de structures conditionnelles que nous le souhaitons ! Attention cependant aux « forêts de ifs ».

Question 4


Que se passe-t-il si la condition d'une structure conditionnelle simple n'est pas remplie ?

- ☐ L'exécution du programme s'arrête. Il faut que la condition soit remplie
- ☒ Le programme passe à la suite des instructions, après la structure conditionnelle
-  Si une condition n'est pas remplie, le monde ne s'arrête pas ! Le programme passe simplement à l'exécution des instructions suivantes.

Question 5

```
1 if (i < 1) {  
2     if (I == 1) {  
3         print(I est inférieur à 1 !')  
4     }  
5 }
```


Qu'est-ce que mon programme va afficher ?

- ☒ Rien du tout !
- ☐ i est inférieure à 1
-  Le programme ne va rien afficher, tout simplement parce que si la condition de la première structure conditionnelle est remplie, à savoir « i est inférieure à 1 », cela veut dire que i ne sera jamais égale à 1 et ne pourra jamais mener à l'exécution du second bloc d'instruction.

Exercice p. 5 Solution n°2


Question 1

La structure conditionnelle alternative permet :

- ☐ De continuer l'exécution du programme
- ☒ De proposer une alternative si la condition `if` n'est pas remplie, et qui ne s'exécutera que dans ce cas de figure
- ☐ De proposer une alternative qui s'exécutera dans tous les cas
-  La structure conditionnelle simple ne permettait pas de proposer une alternative dans le cas où la condition n'est pas remplie. Cela est donc permis par la structure conditionnelle alternative, qui exécutera les instructions de « `else` » uniquement si la condition du « `if` » n'est pas remplie.

Question 2


Si la condition `if` n'est pas remplie, la condition `else` s'exécutera forcément.

- ☒ Vrai
- ☐ Faux
-  C'est tout l'intérêt de ce type de structure ! Si `if` est faux, `else` s'exécutera alors obligatoirement.

Question 3

```
1 var i = 5;
2 if (i = 4) {
3     print("Youpi");
4 } else {
5     print("Youpla !");
6 }
```


Qu'est-ce que mon programme va afficher ?

- ☐ Youpi
- ☒ Youpla
- ☐ Youpi youpla
-  Puisque i est égale à 5, la première condition n'est pas remplie. C'est donc les instructions contenues dans le bloc `else` qui s'exécutent.

Question 4

Puis-je proposer plusieurs alternatives avec une structure conditionnelle simple ?

- ☐ Oui
- ☒ Non


 Non, il faut pour cela utiliser une structure conditionnelle alternative.

Exercice p. 7 Solution n°3

Question 1

Que permet une structure conditionnelle généralisée ?


- ☐ D'appliquer une condition à l'ensemble d'un programme
- ☐ D'appliquer une condition à l'ensemble des variables d'un programme
- ☒ De créer plusieurs évaluations de conditions dans une structure conditionnelle

 La structure conditionnelle généralisée permet de créer plusieurs évaluations de conditions dans une structure conditionnelle.

Question 2

Si une condition d'une structure conditionnelle est vraie :

- ☒ Les conditions suivantes ne seront pas évaluées
- ☐ Toutes les conditions suivantes seront évaluées
- ☐ Seulement la condition else de fin sera également évaluée


 Si une condition d'une structure conditionnelle est vraie, les conditions suivantes ne seront pas évaluées (même si elles sont également vraies !).

Question 3

```
1 Var livre = 'Harry Potter';
2 if (livre == 'Le Seigneur des Anneaux')
3     { print('L'auteur est J. R. R. Tolkien');
4 } else if (livre == 'Dune') {
5     print('L'auteur est Franck Herbert');
6 } else
7 {
8     print('L'auteur est J. K. Rowling');
```

Qu'est-ce que le programme affichera ?

- ☐ L'auteur est J. R. R. Tolkien
- ☐ L'auteur est Franck Herbert
- ☒ L'auteur est J. K. Rowling

 Puisque livre est égale à « *Harry Potter* », les deux premières conditions ne sont pas remplies. C'est donc l'instruction else qui s'exécute.

Question 4

Est-il possible d'utiliser l'instruction switch pour obtenir le même résultat qu'avec le programme de la question précédente ?

- ☒ Oui
- ☐ Non
- ☐ Ça dépend

Q Il est tout à fait possible d'utiliser switch dans ce cas. Cela donnerait :

```
1 switch(livre) {
2     case « Le Seigneur des Anneaux »:
3         print('L'auteur est J. R. R. Tolkien');
4         break;
5     case « Dune »:
6         print('L'auteur est Franck Herbert');
7         break;
8     default:
9         print('L'auteur est J. K. Rowling');
10 }
```

Question 5

Si toutes les conditions alternatives sont remplies, que se passe-t-il ?

- ☒ Seules les instructions de la première condition sont exécutées
- ☐ Toutes les instructions de toutes les conditions sont exécutées
- ☐ Aucune instruction n'est exécutée

Q Si toutes les conditions alternatives sont vraies, seule la première condition est évaluée. Le programme sort ensuite de la structure conditionnelle.

Exercice p. 8 Solution n°4

Question 1

Un opérateur ternaire est :

- ☒ Une structure conditionnelle simplifiée
 - ☐ Une structure conditionnelle permettant des actions supplémentaires
- Q Un opérateur ternaire ne permet que des opérations conditionnelles très simples.

Question 2

Si je souhaite vérifier qu'une expression n'est pas nulle avant d'en utiliser la valeur, quelle syntaxe puis-je utiliser ?

- ☒ `expr1 ? expr1 : expr2`
- ☒ `expr1 ?? expr2`
- ☐ Aucune des deux


Q Les deux syntaxes sont possibles et mèneront au même résultat.

Question 3

```
1 var i = 3 < 4 ? 4 : 5;
```

Quelle valeur prendra i ?

☒ 4☐ 5☐ 3


 « i » prendra la valeur 4, car 3 est bien inférieure à 4 et c'est donc la première expression qui s'applique.

Question 4

```
1 var a = null;
2 var b = a ?? 'Seconde expression'
```

Quelle valeur prendra b ?


☐ null☒ 'Seconde expression'☐ 'a'☐ a

 Puisque a vaut null, b ne prendra pas la valeur de a, mais la valeur de la seconde expression.

Question 5

Je souhaite imbriquer plusieurs structures conditionnelles - qu'est-ce qui est possible ?

☒ `expr1 ? (Expr2 ?? expr3) : expr4`☒ `if (expr1) { if(expr2) {} }`☐ Aucune des deux propositions

 Les deux sont possibles ! Cependant, la première syntaxe est à éviter car elle devient très vite illisible. Il faut privilégier la seconde syntaxe pour imbriquer plusieurs structures conditionnelles.

p. 10 Solution n°5

Cet exercice requiert deux types de structures conditionnelles : des alternatives et une généralisée.

Note : il aurait aussi été possible d'utiliser l'instruction switch.

```
1 import 'dart:io';
2
3 main() {
4   var i = 4;
5
6   if (i % 2 == 0) {
7     print('La variable est un multiple de 2 !');
8     if (i == 2) {
9       print('La variable est égale à 2');
10    } else {
11      print("La variable n'est pas 2 !");
12    }
13  } else if (i % 3 == 0) {
14    print('La variable est un multiple de 3 !');
15
16    if (i == 3) {
```

```

17     print('La variable est égale à 3');
18 } else {
19     print("La variable n'est pas 3!");
20 }
21 } else {
22     print('La variable n'est ni un multiple de 3 ni un multiple de 2 ');
23 }
24 return 0;
25 }

```

Exercice p. 10 Solution n°6

Question 1

Qu'est-ce qui s'affiche lorsque le programme suivant est exécuté ?

```


1 main () {
2
3     var i = 7;
4
5     if (i < 8) {
6         print(« Première condition »);
7     }
8     else if (i == 7) {
9         print(« Seconde condition »);
10    }
11    else {
12        print(« Ne rentre dans aucune condition »);
13    }
14 }

```

☒ Première condition

☐ Seconde condition

☐ Ne rentre dans aucune condition

 La première condition étant vraie, les conditions suivantes ne sont pas exécutées. Ainsi, le programme affichera « Première condition ».


Question 2

Je souhaite créer un programme affichant « Premier cas » si la variable x est égale à 4, mais aussi « Deuxième cas » si la variable est paire. Quelles instructions puis-je utiliser ?

☐ if... else

☐ if... else if

☒ if... if

 Si nous utilisons else ou else if, la deuxième condition ne sera pas évaluée sur la première condition si if est vraie. Or, nous souhaitons que les deux soient évaluées dans tous les cas. Nous pouvons donc utiliser if, puis if à nouveau.


Question 3

Quel est le bon opérateur pour une condition parmi les propositions suivantes ?

☒ ==

☐ =

☐ !=

 « = » est utilisé pour assigner une valeur à une variable. « == », cependant, vérifie l'égalité entre deux éléments, et peut donc être utilisé dans une condition. « != » n'existe pas.

Question 4

Quelle est la bonne version du programme ci-dessous avec l'utilisation de l'instruction switch ?

```
1 var x = 10;
2 if (x == 5) {
3     print(« x est égale à 5 »);
4 }
5 else if (x == 8) {
6     print(« x est égale à 8' »);
7 }
8 else if (x == 10) {
9     print(« x est égale à 10 »);
10 }
11 else {
12     print(« x ne rentre dans aucune des conditions »);
13 }
```

```
1 Switch (x) {
2     case 5:
3         print(« x est égale à 5 »);
4         break;
5     case 8:
6         print(« x est égale à 8' »);
7         break;
8     case 10:
9         print(« x est égale à 10 »);
10        break;
11    default:
12        print(« x ne rentre dans aucune des conditions »);
13 }
```

```
1 Switch (x) {
2     case 5:
3         print(« x est égale à 5 »);
4     case 8:
5         print(« x est égale à 8' »);
6     case 10:
7         print(« x est égale à 10 »);
8     default:
9         print(« x ne rentre dans aucune des conditions »);
10 }
```

☐ La première

☐ La deuxième

☐ La troisième

☒ Les deux premières

☐ Les trois

Q Dans ce cas de figure, il est impossible que deux conditions soient vraies. Il n'est donc pas obligatoire de rajouter l'instruction break pour sortir de l'instruction switch. Les deux premières fonctionnent !

Question 5

Puis-je imbriquer des structures conditionnelles dans une structure conditionnelle généralisée ?

- ☒ Oui
- ☐ Non
- ☐ Pas toujours

Q Il est tout à fait possible d'imbriquer tout type de structures conditionnelles, et même de mélanger les types.