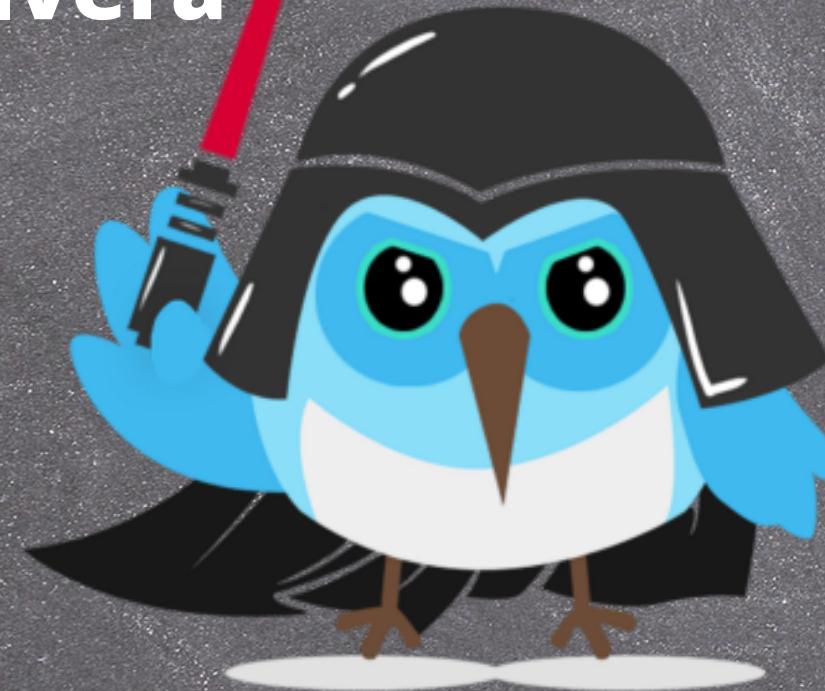


David Silvera



COME TO THE
DART
—SIDE



- Langage de programmation
- Applications mobiles, de bureau, de serveur et web

David Silvera



Flutter

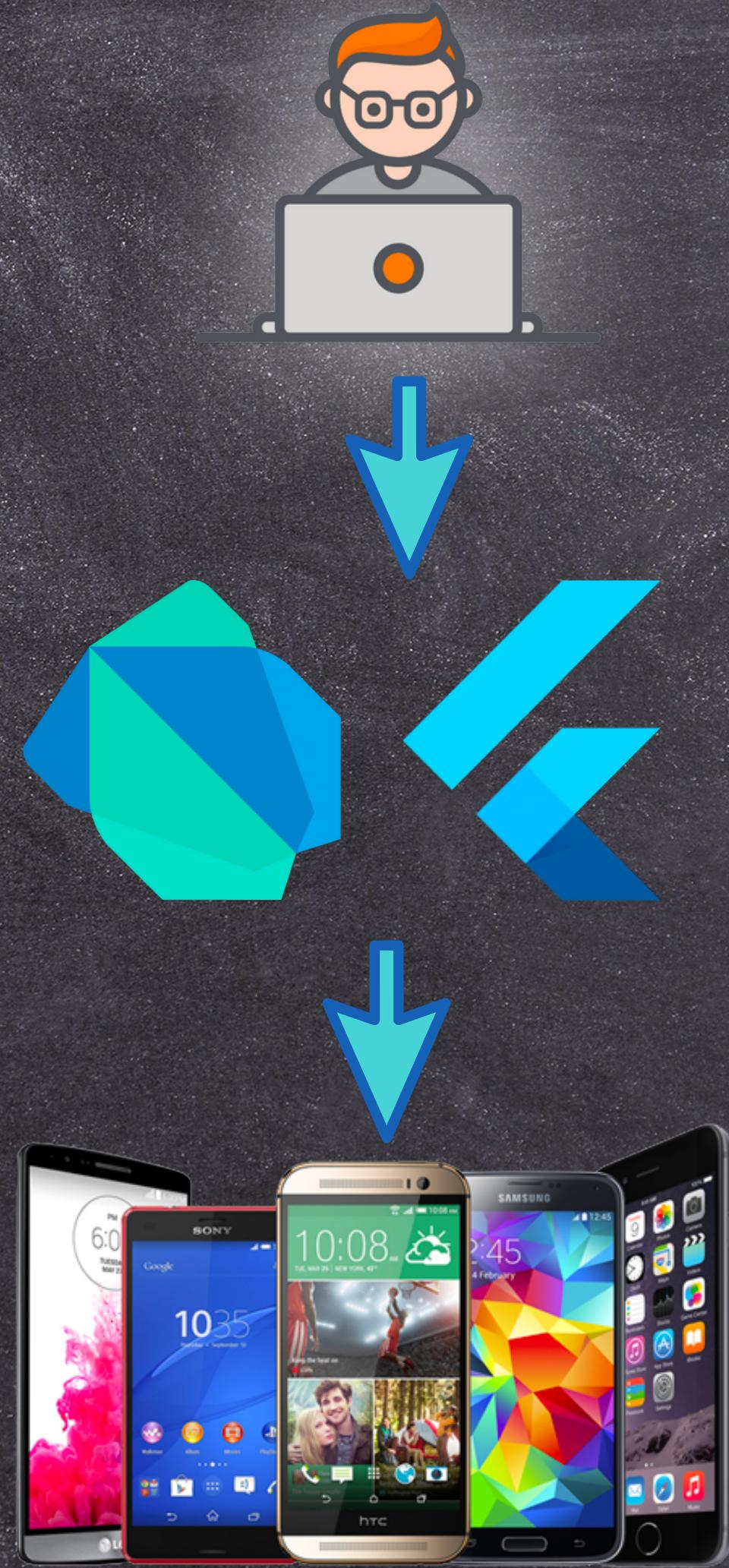


- Kit de développement logiciel
- 1ère version stable : 4 décembre 2018
- 1ère version avec Web et Desktop : 3 mars 2021

mars 2021



David Silvera





David Silvera



<https://dashatar-dev.web.app>



David Silvera

AFFICHER UNE PHRASE DANS LA CONSOLE

```
print("C'est parti!");
```



David Silvera

AJOUTER UN COMMENTAIRE

```
/*
```

```
* Ceci est un long commentaire  
* sur plusieurs lignes
```

```
*/
```

```
// Ceci est un commentaire sur 1 ligne
```



David Silvera

LES VARIABLES PERMETTENT DE STOCKER UNE DONNÉE EN MÉMOIRE

- un nom
- un indicateur pour savoir si je l'ai testée
- un nombre de personne
- des quantités d'ingrédient



David Silvera

LES VARIABLES PERMETTENT DE STOCKER UNE DONNÉE EN MÉMOIRE

- un nom (chaine de caractère → String)

- un indicateur pour savoir si je l'ai
testée (vrai ou faux → Booléen)

- un nombre de personne (un nombre →
Integer)

- des quantités d'ingrédient (nombre à
virgule → Double)





David Silvera

LES VARIABLES

PERMETTENT DE STOCKER UNE DONNÉE EN MÉMOIRE

```
String title = "Gateau au choco";
```

```
int nbPerson = 8;
```

```
bool isLiked = false;
```

```
double chocoGramme = 250.6;
```

```
var title = "Gateau au choco";
```

```
var nbPerson = 8;
```

```
var isLiked = false;
```

```
var chocoGramme = 250.6;
```

David Silvera

LES LISTES

STOCKENT PLUSIEURS VARIABLES DU MÊME TYPE

Une liste d'ingrédient



de chocolat pâtissier

de farine

de sucre en poudre

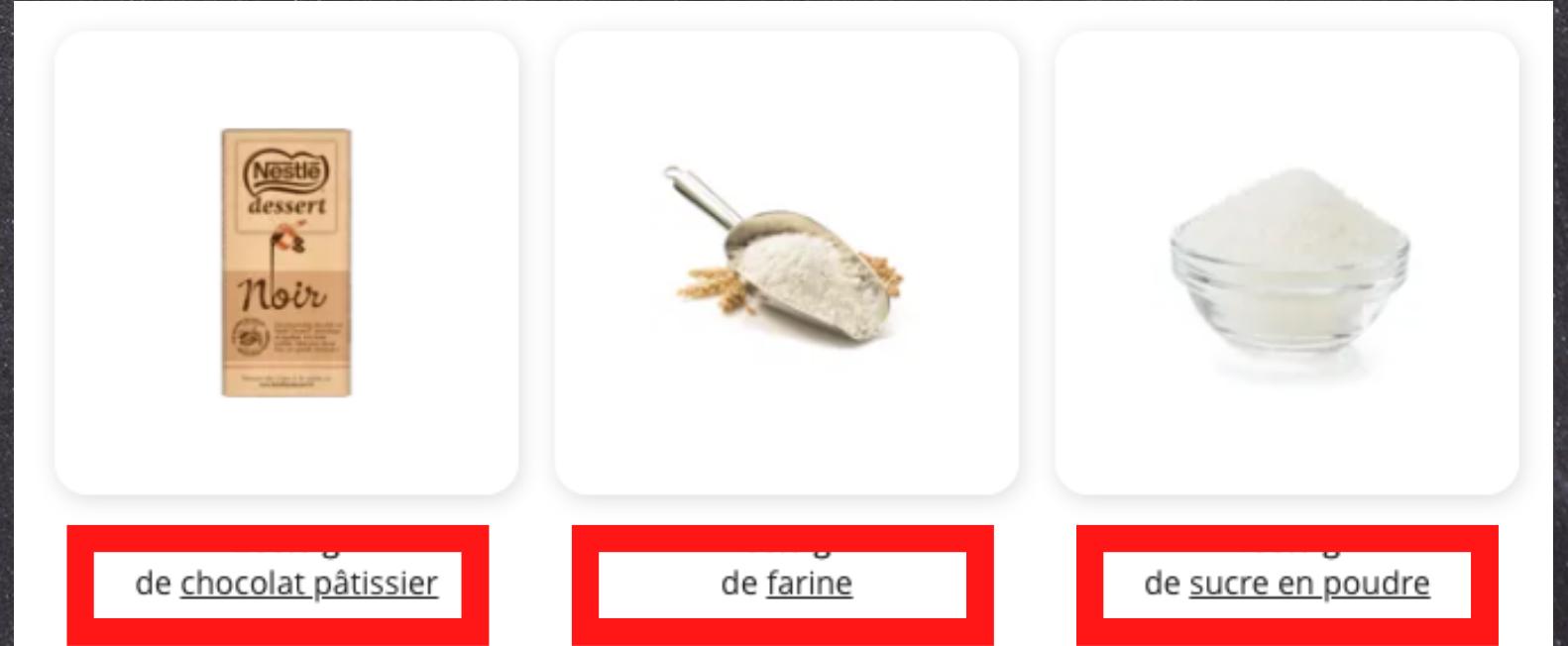


David Silvera

LES LISTES STOCKENT PLUSIEURS VARIABLES DU MÊME TYPE

```
List<String> ingredients = [];  
ingredients.add("Chocolat");  
ingredients.add("Farine");  
ingredients.add("Sucre");  
print(ingredients);
```

Une liste d'ingrédient





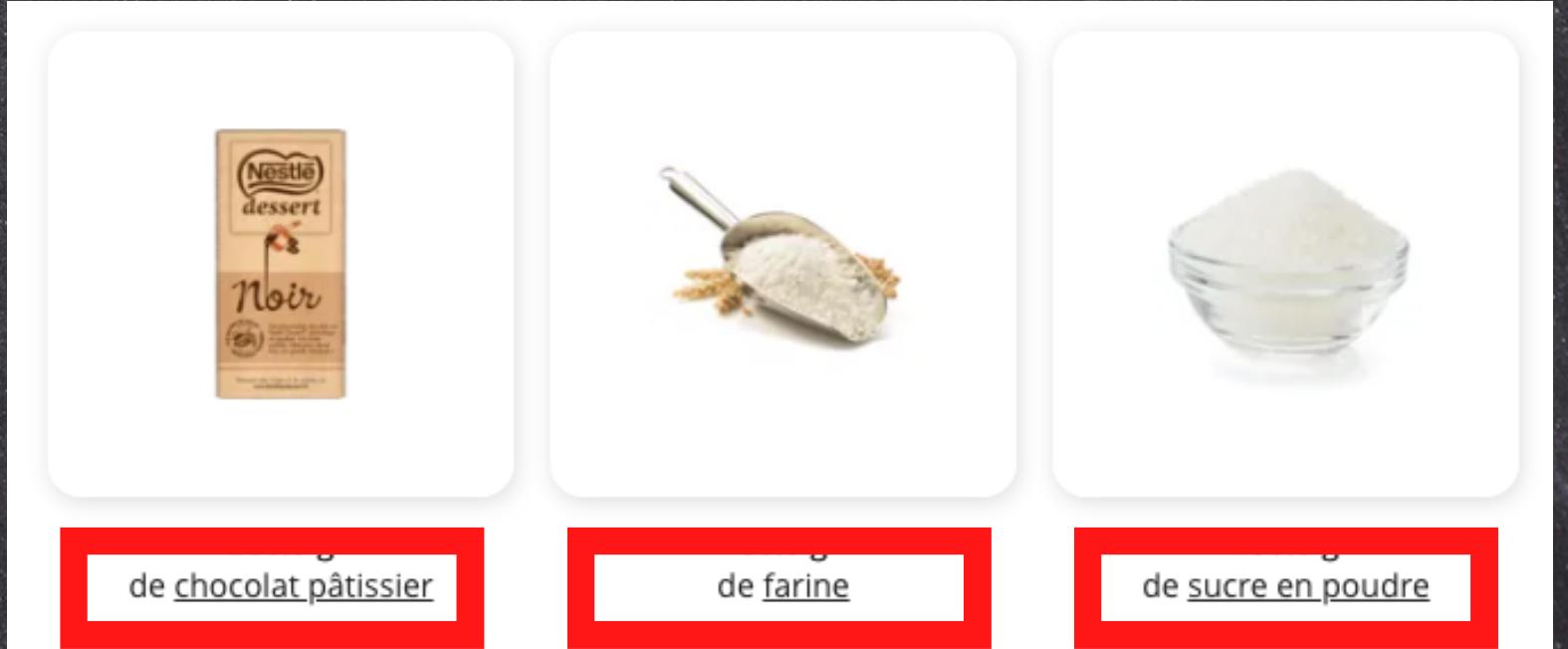
David Silvera

LES SETS

STOCKENT PLUSIEURS VARIABLES DU MÊME TYPE SANS DOUBLON

```
Set<String> ingredients = Set();  
ingredients.add("Chocolat");  
ingredients.add("Farine");  
ingredients.add("Sucre");  
print(ingredients);
```

Une liste d'ingrédient





David Silvera

LES HASHMAP STOCKENT UN ENSEMBLE DE CLÉ / VALEUR

Une liste d'ingrédient avec sa quantité





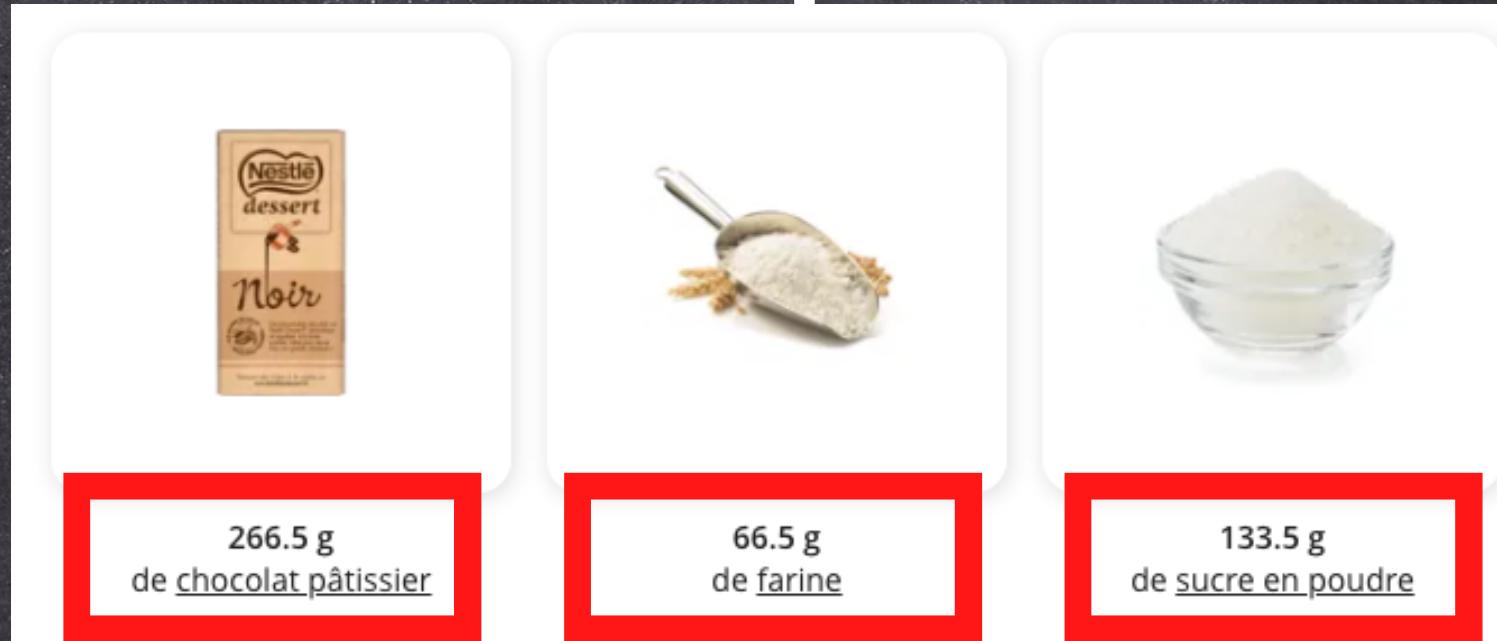
David Silvera

LES HASHMAP STOCKENT UN ENSEMBLE DE CLÉ / VALEUR

```
Map<String, double> ingredients =  
{  
    "Chocolat" : 266.5,  
    "Farine" : 66.5,  
    "Sucre" : 133.5  
};
```

```
Map<String, double> ingredients = Map();  
ingredients["Chocolat"] = 266.5;  
ingredients["Farine"] = 66.5;  
ingredients["Sucre"] = 133.5;
```

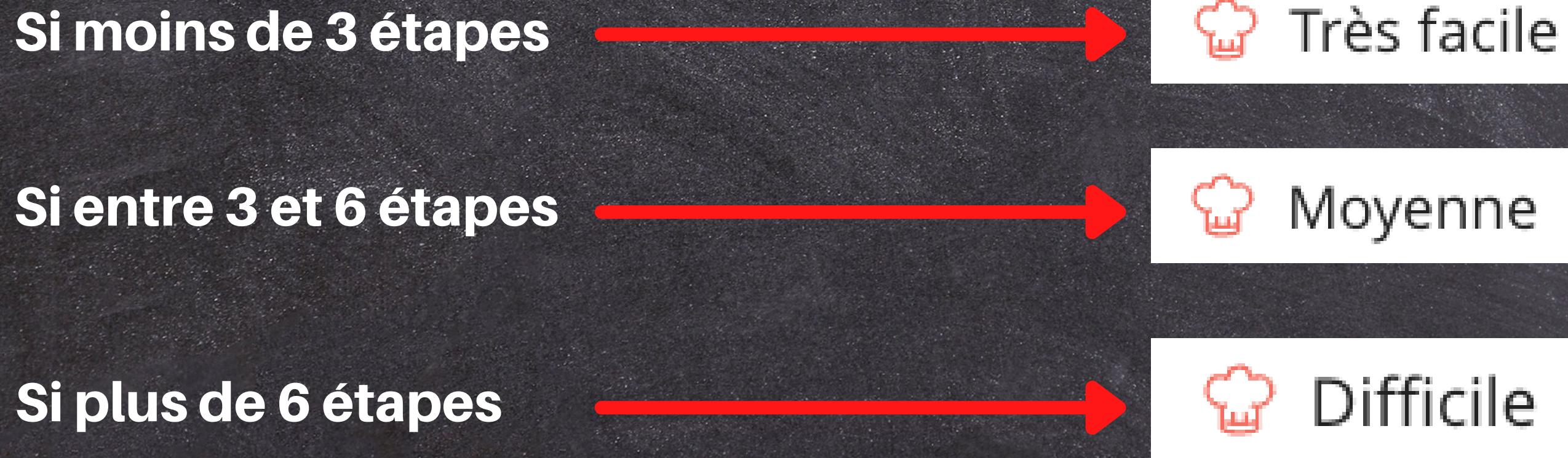
Une liste d'ingrédient
avec sa quantité





David Silvera

LES CONDITIONS





David Silvera

LES CONDITIONS

```
if (nbStep <= 3)
```



 Très facile

```
else if (nbStep > 3 && nbStep <= 6)
```



 Moyenne

```
else if (nbStep > 6)
```



 Difficile



David Silvera

LES CONDITIONS

```
switch (nbStep) {  
    case 1 : {  
        print("Facile");  
        break;  
    }  
    case 2 : {  
        print("Moyenne");  
        break;  
    }  
    default : {  
        print("Diffile");  
        break;  
    }  
}
```



 Très facile

 Moyenne

 Difficile



David Silvera

LES BOUCLES PARCOURIR UN ENSEMBLE



200 g
de chocolat
pâtissier



50 g
de farine



100 g
de sucre en
poudre



100 g
de beurre + une
noix pour le moule



3
oeufs



David Silvera

LES BOUCLES

PARCOURIR UN ENSEMBLE

```
var ingredients = ["chocolat", "farine", "sucre", "beurre", "oeuf"];  
for (int i = 0; i < ingredients.length; i++) {  
    print("${i+1} - ${ingredients[i]}");  
}  
-----  
for (var ingredient in ingredients) {  
    print(ingredient);  
}
```

David Silvera

LES BOUCLES

PARCOURIR UN ENSEMBLE





David Silvera

LES BOUCLES PARCOURIR UN ENSEMBLE

```
var ingredients = ["chocolat", "farine", "sucre", "beurre", "oeuf"];  
  
var i = 0;  
  
while (i < ingredients.length && ingredients[i] != "beurre") {  
    print(ingredients[i]);  
    i++;  
}
```



3
oeufs



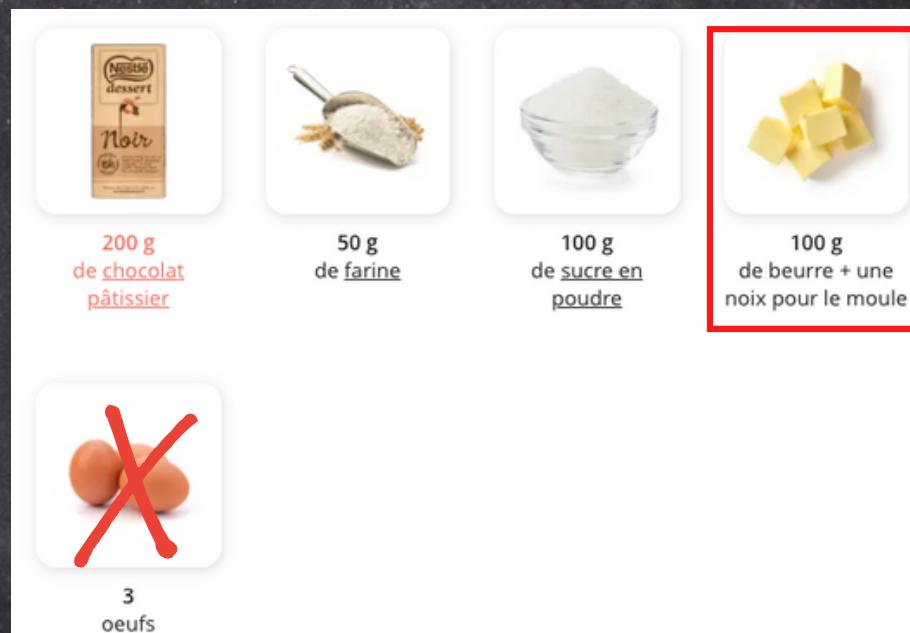
David Silvera

LES BOUCLES

PARCOURIR UN ENSEMBLE

```
var ingredients = ["chocolat", "farine", "sucre", "beurre", "oeuf"];
var i = 0;
do {
    print(ingredients[i]);
    i++;
} while (i < ingredients.length && ingredients[i] != "beurre");
```

=> EXECUTE AU MOINS UNE ITÉRATION





David Silvera

LES FONCTIONS

ACTIONS À EXECUTER

- Afficher les images de la recettes
- Afficher la liste des ingrédients
- Afficher les étapes
- Ajouter aux favoris
- Partager la recette



David Silvera

LES FONCTIONS

ACTIONS À EXECUTER

type de retour de la fonction



nom de la fonction



Paramètres de la fonction

```
String getSteps(int nbStep, String prefix) {  
    String result = "";  
  
    for (int i = 0; i < nbStep; i++) {  
        result += "$prefix ${i + 1}\n";  
    }  
  
    return result;  
}
```

Retour de la fonction

A circular profile picture of a man with glasses and a beard, wearing a blue shirt with a red play button logo.

David Silvera

LES FONCTIONS

ACTIONS À EXECUTER

=> Paramètres nommés

```
String getSteps({required int nbStep, required String prefix}) {  
    String result = "";  
    for (int i = 0; i < nbStep; i++) {  
        result += "$prefix ${i + 1}\n";  
    }  
    return result;  
}
```

A circular profile picture of a man with glasses and a beard, wearing a blue shirt with a red play button logo.

David Silvera

LES FONCTIONS

ACTIONS À EXECUTER

=> Valeur par défaut

```
String getSteps({required int nbStep, String prefix = "Etape"}) {  
    String result = "";  
    for (int i = 0; i < nbStep; i++) {  
        result += "$prefix ${i + 1}\n";  
    }  
    return result;  
}
```

A circular profile picture of a man with glasses and a beard, wearing a blue shirt with a red play button logo.

David Silvera

LES FONCTIONS

ACTIONS À EXECUTER

=> Valeur par défaut avec variable non nommée

```
String getSteps(int nbStep, [String prefix = "Etape"]) {  
    String result = "";  
    for (int i = 0; i < nbStep; i++) {  
        result += "$prefix ${i + 1}\n";  
    }  
    return result;  
}
```



David Silvera

LES FONCTIONS SHORT ACTIONS À EXECUTER

```
int getNbSteps(int nbIngredient) => nbIngredient * 2;
```

David Silvera

LES CLASSES

```
void main() {  
    Dessert recipe = Dessert(name : "Gateau au choco", ingredients : ["oeuf", "farine", "sucre", "levure"], steps : ["Step 1", "Step 2"]);  
    print(recipe.type);  
}
```

```
class Recipe {
```

```
    String name;  
    List<String> ingredients;  
    List<String> steps;  
    String type;
```

```
    Recipe({required this.name, required this.ingredients, required this.steps, required this.type});
```

Nom de la classe

Variables de la classe

```
    String formatIngredients() {  
        String result = "";  
        for (int i = 0; i < ingredients.length; i++) {  
            var isLast = i != ingredients.length -1;  
            result += ingredients[i] + (isLast ? "," : "");  
        }  
  
        return result;  
    }
```

Constructeur de la classe

```
}
```

Héritage

```
class Dessert extends Recipe {  
    Dessert({required String name, required List<String> ingredients, required List<String> steps})  
        : super(name : name, ingredients : ingredients, steps : steps, type: "Dessert");  
}
```

Nom de la classe héritée



**David Silvera
Sur YouTube**