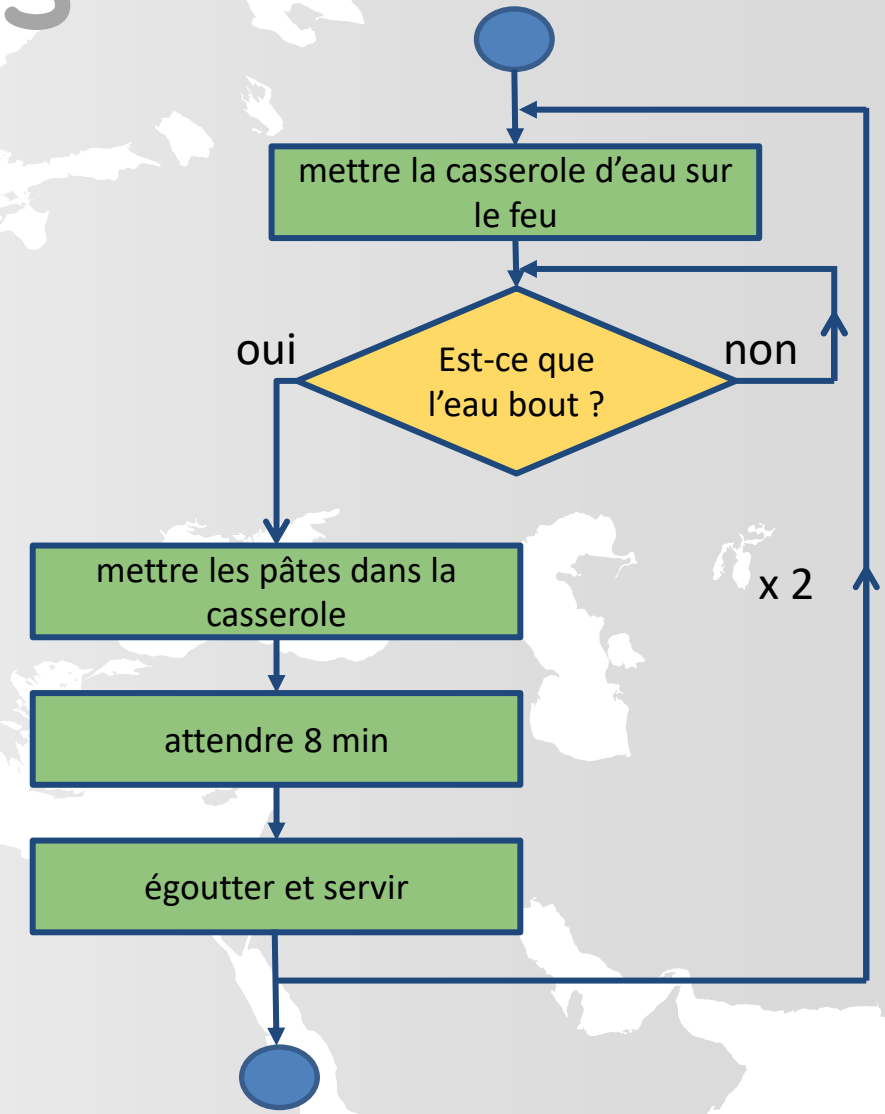
App Inventor basics

Qu'est-ce qu'un Algorithme ?



Cette présentation est en partie inspirée et issue de celle du Pr Ralph Morelli dans le cours Mobile CSP

http://onvaessayer.org Marie-Caroline Gries, Pierre Huguet

Qu'est-ce qu'un algorithme ?

Définition du Larousse

- Un algorithme est un ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un ordinateur.

Qu'est-ce qu'un algorithme ?

Définition du Larousse

- Un algorithme est un ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un ordinateur.

Définition Wikipedia

- Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat

Qu'est-ce qu'un algorithme ?

Définition du Larousse

- Un algorithme est un ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un ordinateur.

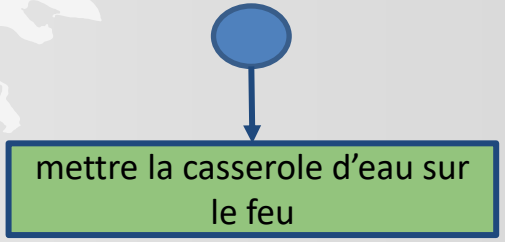
Définition Wikipedia

- Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat

Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Qu'est-ce qu'un algorithme ?



Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

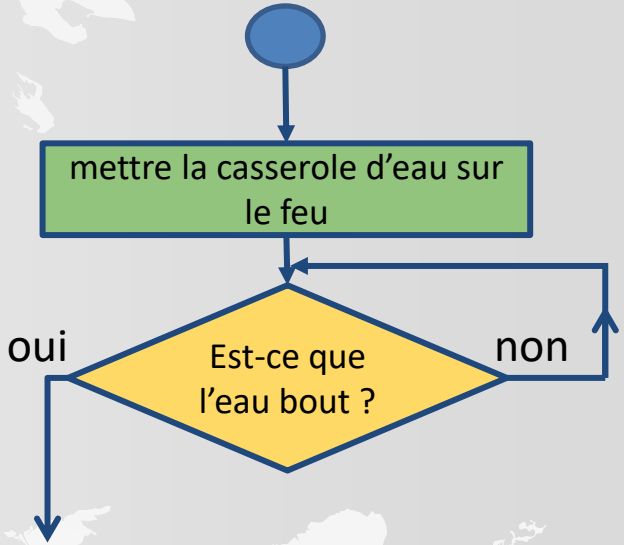
- mettre la casserole sur le feu.

Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer

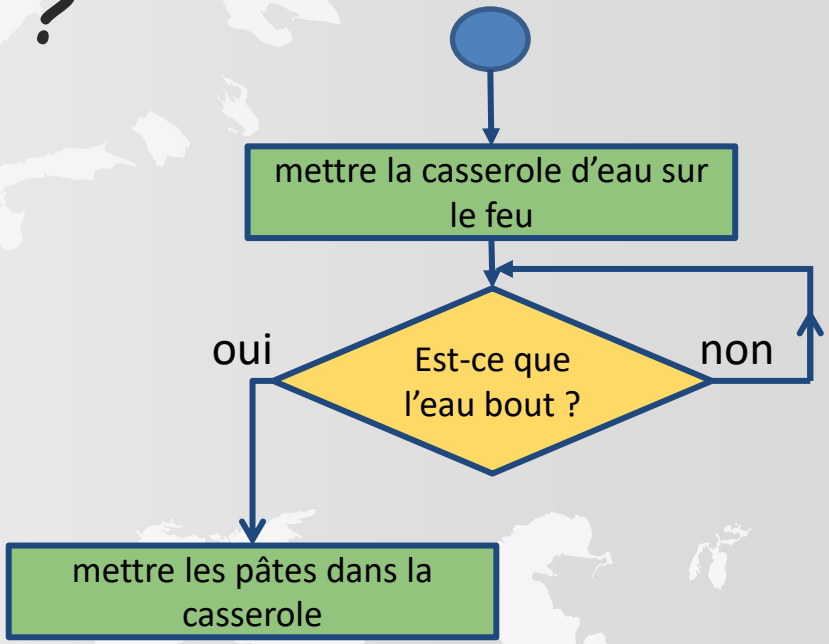


Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer
- mettre les pates dans la casserole

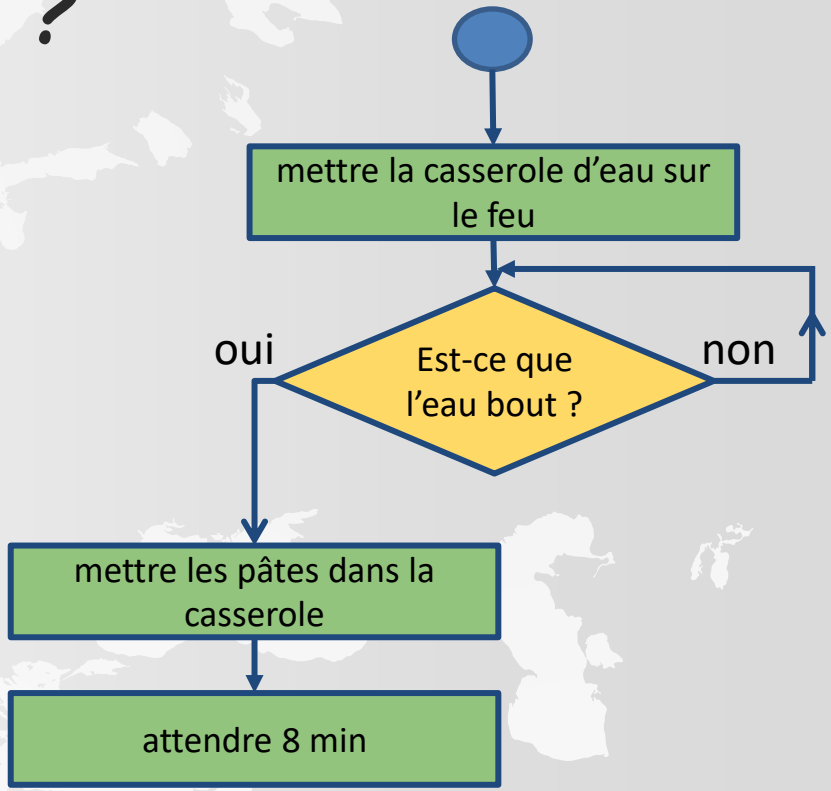


Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer
- mettre les pates dans la casserole
- attendre 8 mn

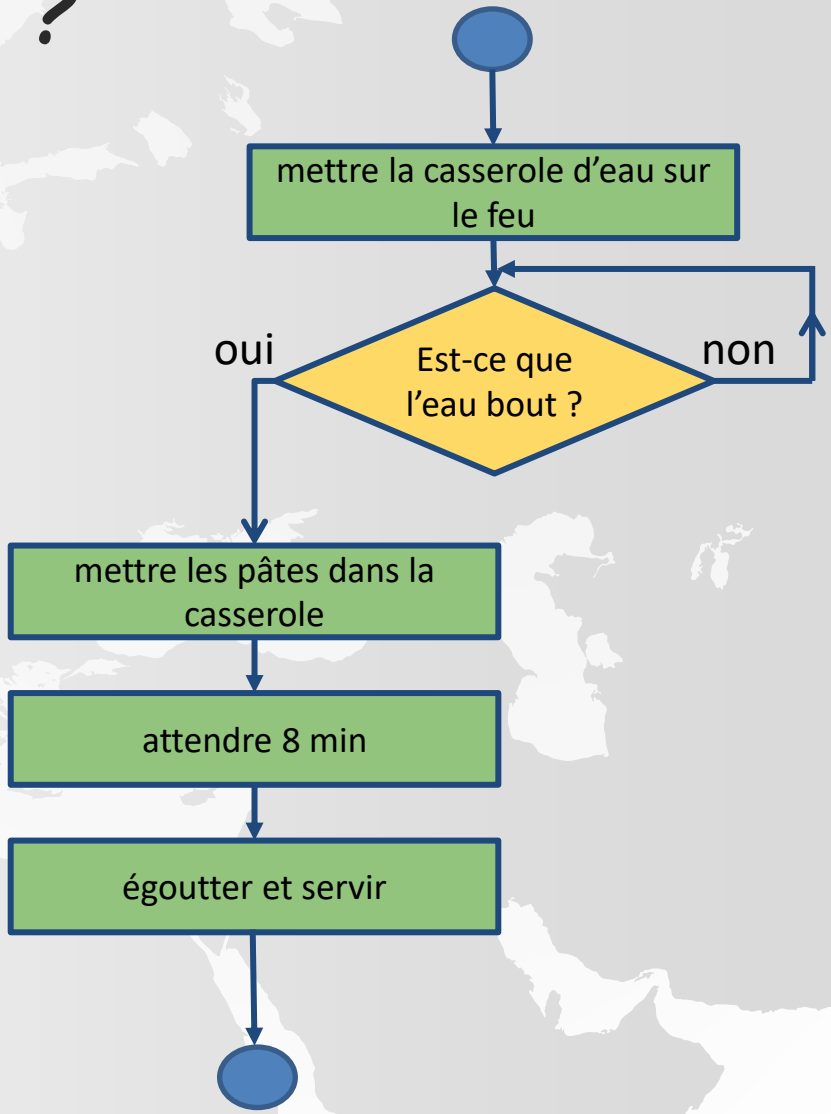


Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer
- mettre les pates dans la casserole
- attendre 8 mn
- égoutter et servir

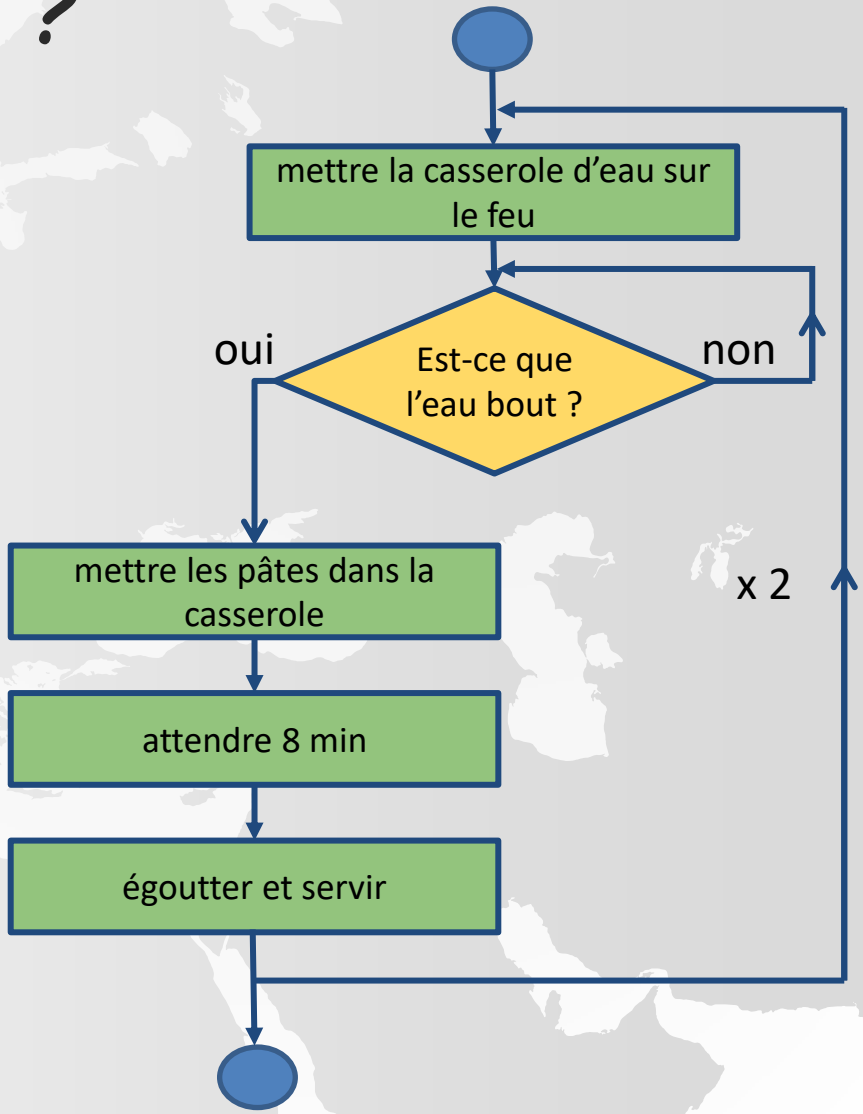


Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer
- mettre les pates dans la casserole
- attendre 8 mn
- égoutter et servir
- (répéter si nécessaire)



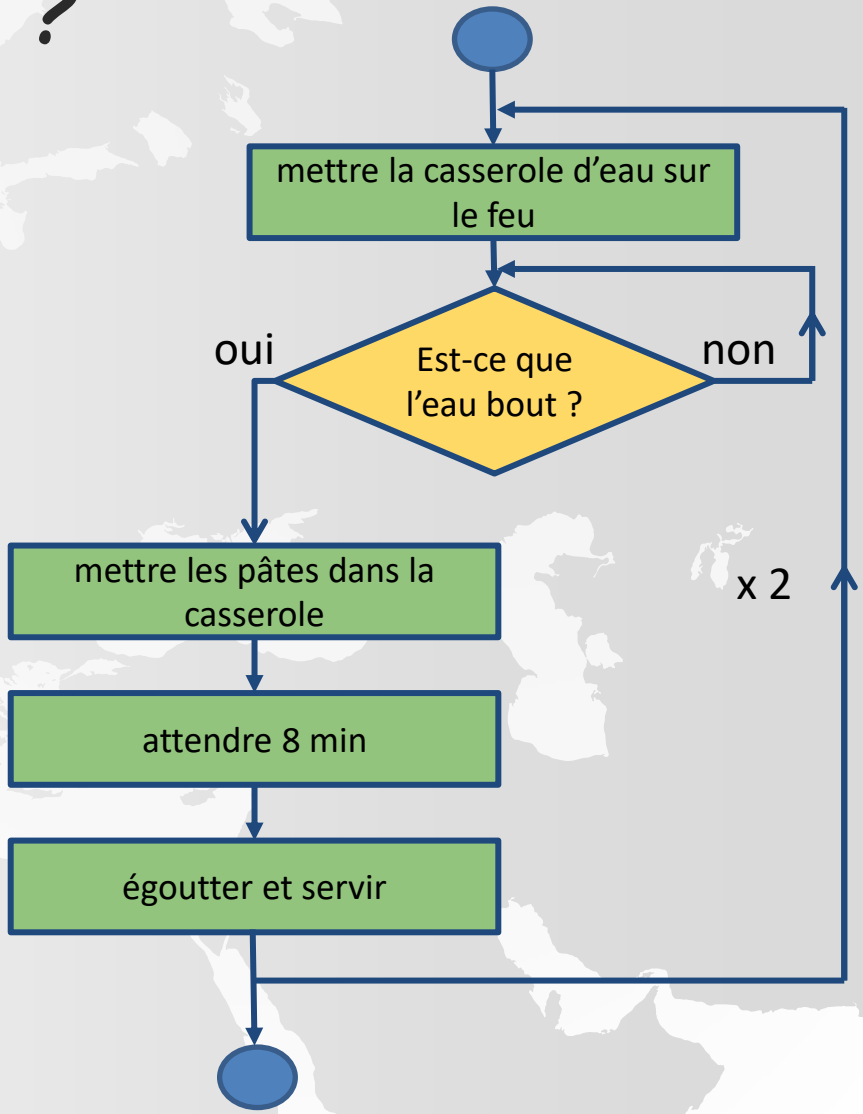
Qu'est-ce qu'un algorithme ?

Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer
- mettre les pates dans la casserole
- attendre 8 mn
- égoutter et servir

La Différence ?



Qu'est-ce qu'un algorithme ?

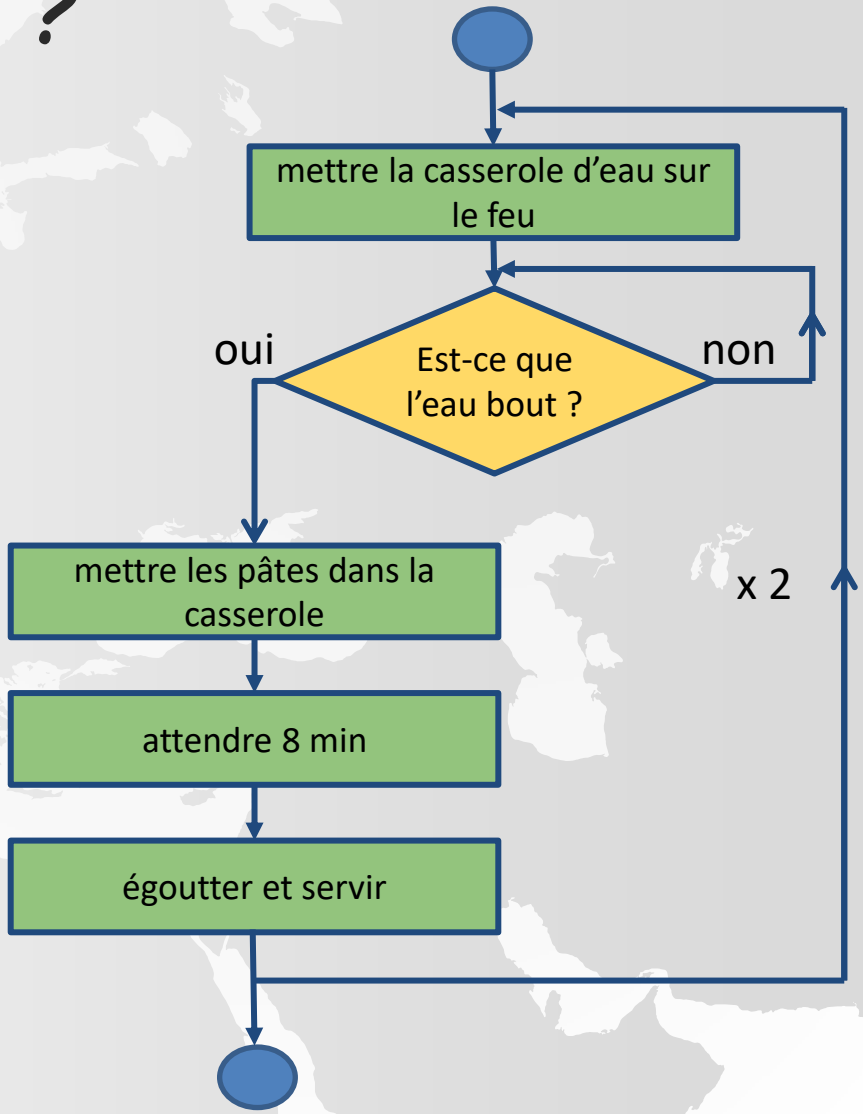
Un *algorithme* est similaire à une *recette*.

Pour cuire des pates :

- mettre la casserole sur le feu.
- est-ce que l'eau bout ?
  - Si non : vérifier à nouveau
  - Si oui : continuer
- mettre les pates dans la casserole
- attendre 8 mn
- égoutter et servir

La Différence ?

Les algorithmes informatiques sont plus précis.



Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va au supermarché, prends une boite de lentilles,
si ils ont des œufs, prends en douze."

Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va au supermarché, prends une boite de lentilles,
si ils ont des œufs, prends en douze."

et Chéri revient avec …

Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va au supermarché, prends une boite de lentilles,
si ils ont des œufs, prends en douze."

et Chéri revient avec 12 boites de lentilles.

Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va au supermarché, prends une boite de lentilles,
si ils ont des œufs, prends en douze."

et Chéri revient avec 12 boites de lentilles.

Problème ?

Chéri a fait ce qui lui a été demandé …
mais une douzaine de quoi ?
l'algorithme est ambigu, non ?

Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va chercher une boite de lentilles au supermarché et tant que tu y es prends des œufs."

et Chéri …

Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va chercher une boite de lentilles au supermarché et tant que tu y es prends des œufs."

et Chéri n'est jamais revenu …

Un algorithme doit être précis

Attention à l'ambiguïté :

"Chéri, va chercher une boite de lentilles au supermarché et tant que tu y es prends des oeufs."

et Chéri n'est jamais revenu …

- *Problème ?*

  Chéri a fait ce qui lui a été demandé …
  "tant qu'il y était" il a continué à prendre des œufs …
  Boucle infinie …

Points clefs sur les algorithmes

Un algorithme doit être précis.

Un algorithme est défini pas-à pas (séquence) :

Chaque pas ou étape doit être précis et faisable.

Points clefs sur les algorithmes

Un algorithme doit être précis.

Un algorithme est défini pas-à pas (séquence) :

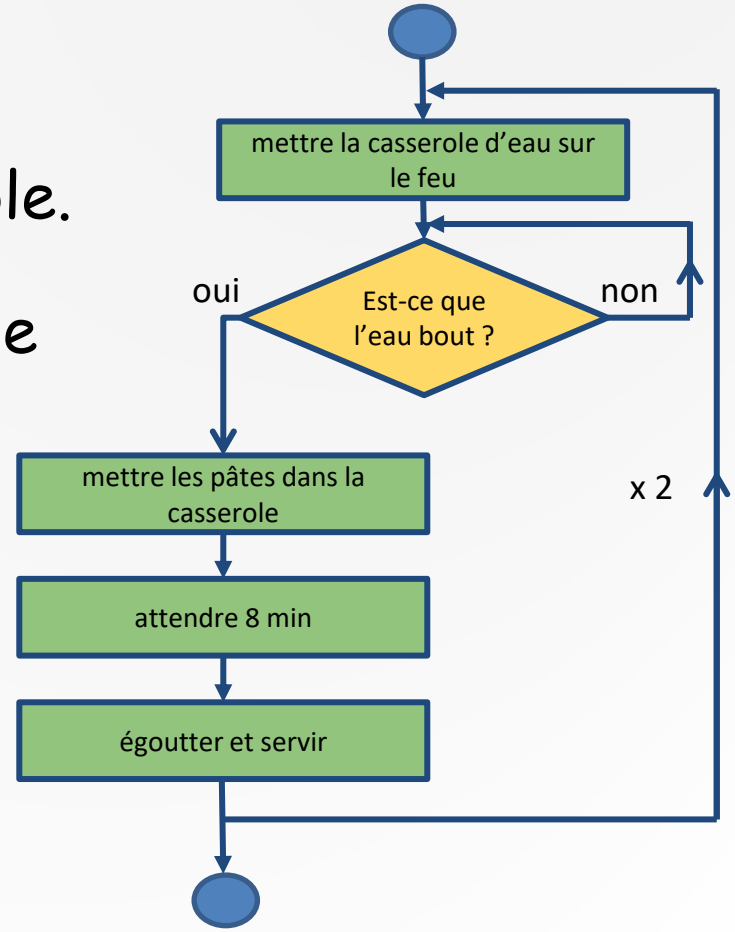
Chaque pas ou étape doit être précis et faisable.

Tous les algorithmes peuvent être réalisés à partir de

séquences : suite ordonnée d'instructions

sélections : tests, si … sinon …

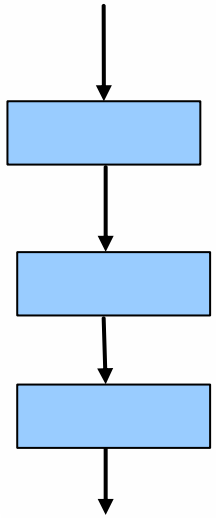
répétitions : boucles



Base des algorithmes

Séquence, Sélection, Répétition

la *séquence*, la *sélection*, et la *répétition*.

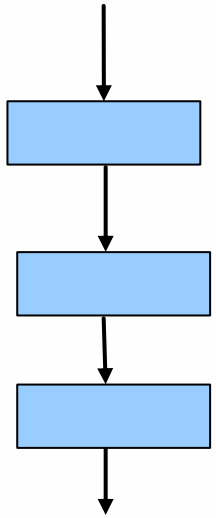


Séquence

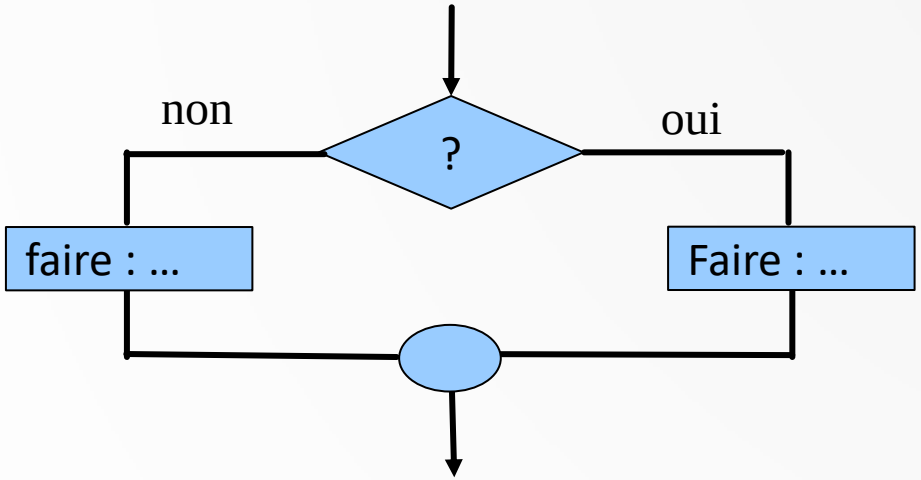
Base des algorithmes

Séquence, Sélection, Répétition

la *séquence*, la *sélection*, et la *répétition*.



Séquence

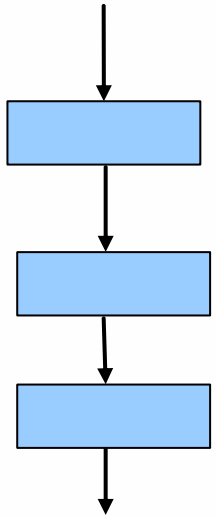


Sélection
(Branchement)

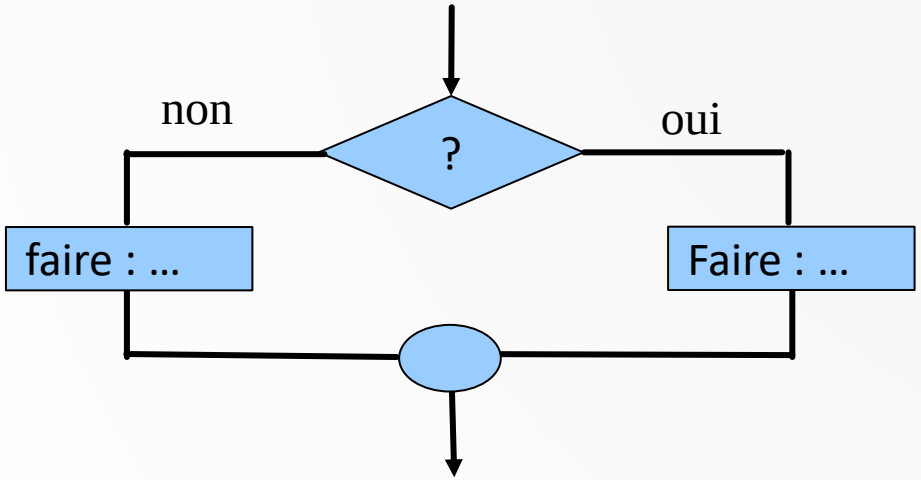
Base des algorithmes

Séquence, Sélection, Répétition

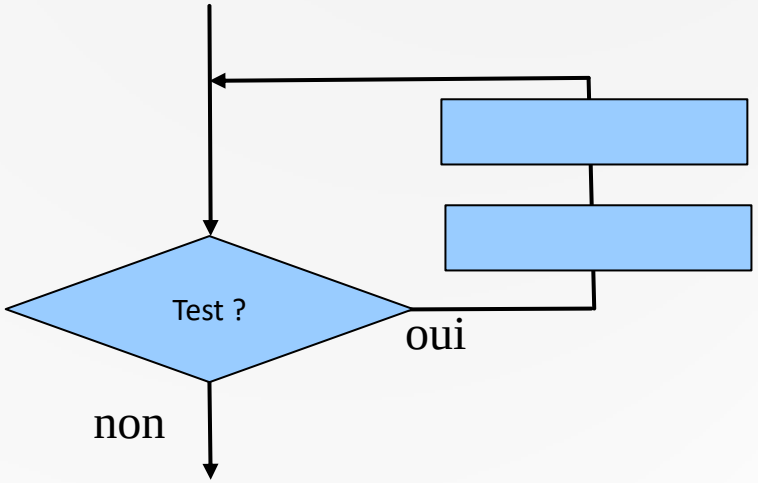
la *séquence*, la *sélection*, et la *répétition*.



Séquence



Sélection
(Branchement)



Répétition
(Bouclage)

Séquence

Une *Séquence* c'est une suite d'instructions que l'ordinateur exécute une par une dans l'ordre où elles sont écrites

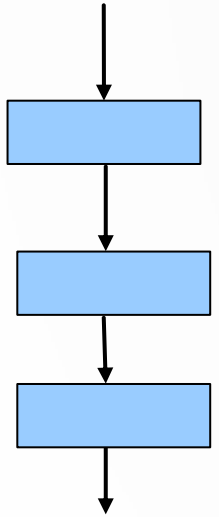


Diagramme d'une Séquence – chaque rectangle correspond à une instruction.

Algorithme de calcul de la moyenne de 70, 80, et 90.

Sélection

La *Sélection* utilise une condition booléenne (vraie ou fausse) pour décider de la suite ou de la branche des instructions à exécuter.

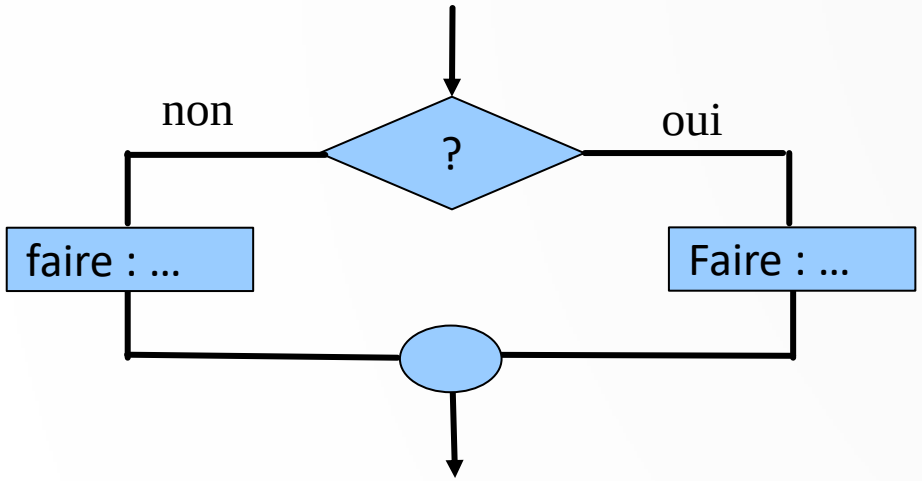
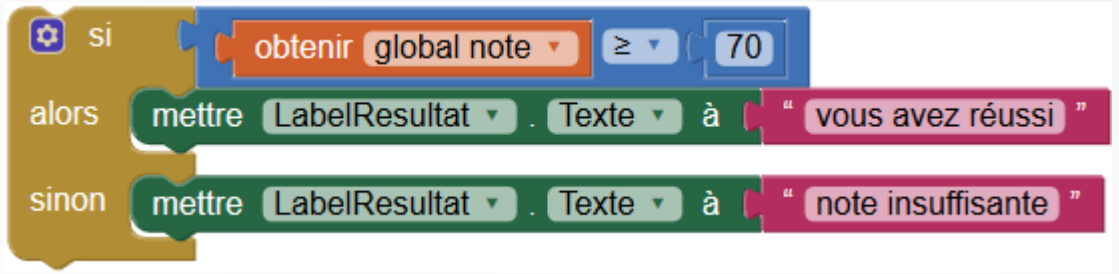


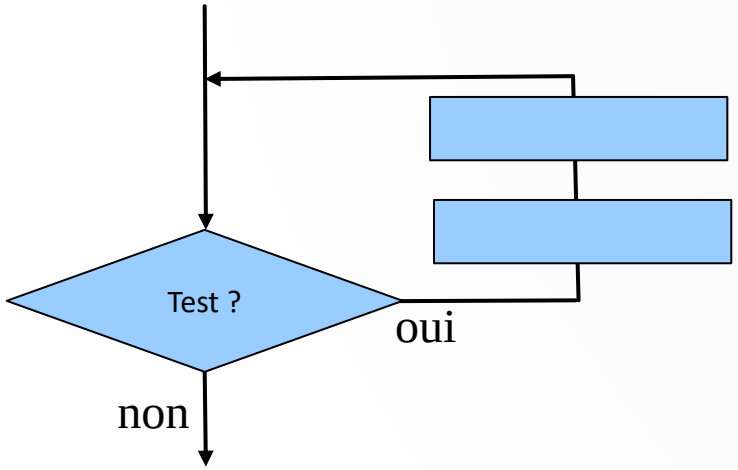
Diagramme de sélection -- le losange exprime une condition; le cercle un connecteur.

Algorithme pour décider si "note" est une note suffisante.



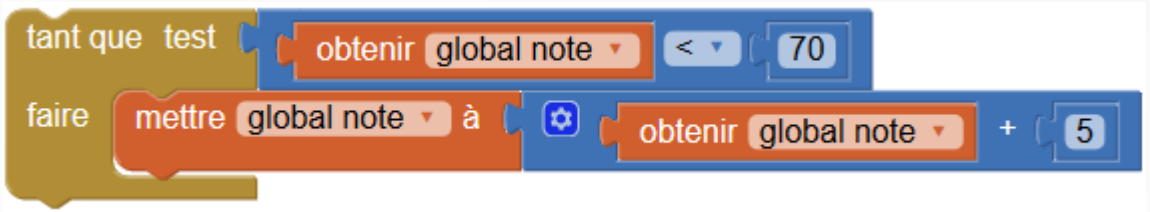
Répétition

La *répétition* (ou *itération*) c'est la répétition de la partie d'un algorithme. On peut préciser le nombre de fois ou effectuer la répétition tant qu'une condition est remplie.



Répétition
(Boucle)

algorithme qui ajoute 5 points à la note jusqu'à ce qu'elle atteigne ou dépasse 70.



Points clefs sur les algorithmes

Un algorithme doit être précis.

Un algorithme est défini pas-à pas (séquence) :

Chaque pas ou étape doit être précis et faisable.

Tous les algorithmes peuvent être réalisés à partir de

séquences : suite ordonnée d'instructions

sélections : tests, si … sinon …

répétitions : boucles

Points clefs sur les algorithmes

Un algorithme doit être précis.

Un algorithme est défini pas-à pas (séquence) :
Chaque pas ou étape doit être précis et faisable.

Tous les algorithmes peuvent être réalisés à partir de
séquences : suite ordonnée d'instructions
sélections : tests, si … sinon …
répétitions : boucles

Les algorithmes peuvent être exprimés avec
le langage naturel,
le pseudocode et les diagrammes fonctionnels
un langage de programmation.

Formulation des Algorithmes

De nombreux langages permettent d'écrire les algorithmes :

Le langage naturel, *le pseudocode* et les blocs App Inventor.

Français

```
Si la moyenne atteint ou dépasse 70,
afficher "vous avez réussi" sinon,
afficher "note insuffisante"
```

Formulation des Algorithmes

De nombreux langages permettent d'écrire les algorithmes :

Le langage naturel, *le pseudocode* et les blocs App Inventor.

Français

```
Si la moyenne atteint ou dépasse 70,
afficher "vous avez réussi" sinon,
afficher "note insuffisante"
```

Pseudocode :
moins formel qu'un langage de programmation,
mais plus précis que le langage naturel .

```
si moyenne >= 70:
  mettre Label.Text à "vous avez réussi"
sinon:
  mettre Label.Text à "note insuffisante
```

Formulation des Algorithmes

De nombreux langages permettent d'écrire les algorithmes :

Le langage naturel, *le pseudocode* et les blocs App Inventor.

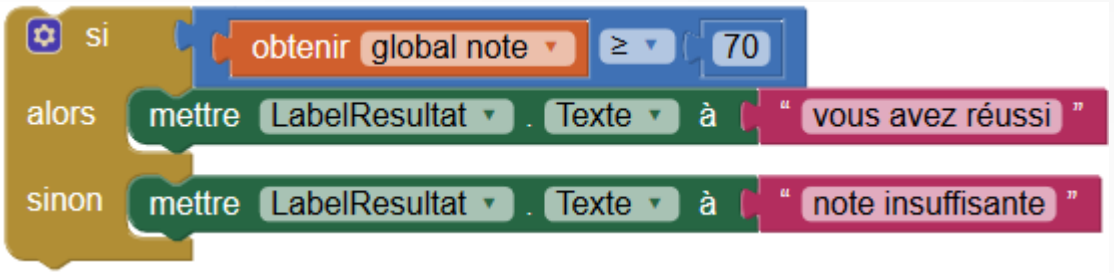
Français

```
Si la moyenne atteint ou dépasse 70,
afficher "vous avez réussi" sinon,
afficher "note insuffisante"
```

Pseudocode :
moins formel qu'un langage de programmation,
mais plus précis que le langage naturel .

```
si moyenne >= 70:
  mettre Label.Text à "vous avez réussi"
sinon:
  mettre Label.Text à "note insuffisante
```

App Inventor



Points clefs sur les algorithmes

Un algorithme est une procédure définie pas-à pas pour résoudre un problème ou obtenir un résultat.

Un algorithme doit être précis : sans ambiguïté

Chaque pas ou étape doit être faisable.

Tous les algorithmes peuvent être réalisés à partir de séquences, de sélections et de répétitions.

Les algorithmes peuvent être exprimés par des diagrammes fonctionnels, du pseudocode, ou de nombreux langages de programmation.

Des algorithmes tous les jours …