Labo d'introduction à l'informatique

pour les mathématiques

Yann Thorimbert

UNIVERSITÉ DE GENÈVE
CENTRE UNIVERSITAIRE D'INFORMATIQUE

Semaine 3
Expressions conditionnelles

Qu’est-ce qu’une condition ?

- Permet d’**adapter** l’exécution du code à un test écrit par la ou le programmeur.

- Ingrédient qui nous manquait jusqu’alors pour traiter **différents cas possibles**.
  Exemples :
  - Gestion des différents cas de Δ dans la formule de Viète.
  - Gestion des malentendus avec l’utilisateur.

Exemple de base | **Utilisation du mot-clé *if***

```
age = input("Quel est l'âge d'Alice ?");
if age < 18
    disp("Alice est mineure.")
end
```

Exemple de base | **Concept de bloc**

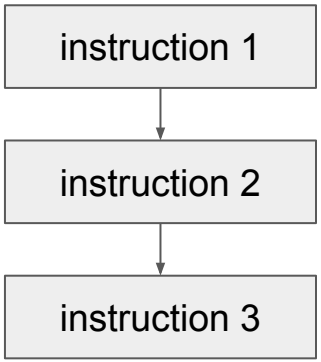
```
age = input("Quel est l'âge d'Alice ?");
disp(“Ceci s’affiche toujours.”)
if age < 18
    disp("Ceci s’affiche uniquement si Alice est mineure.")
end
disp(“Ceci s’affiche toujours également.”)
```

Les conditions comme structure de contrôle

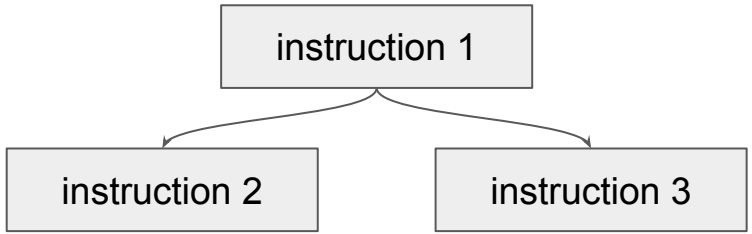
- Une condition est une **structure de contrôle** qui permet d'effectuer des sélections (ou branchements) dans le **flux d'exécution**.

- Le flux d'exécution est **séquentiel** en l'absence de structure de contrôle.

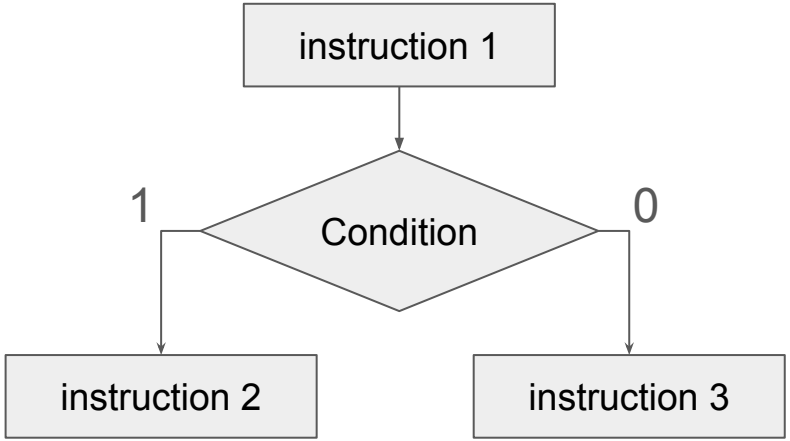
Séquence



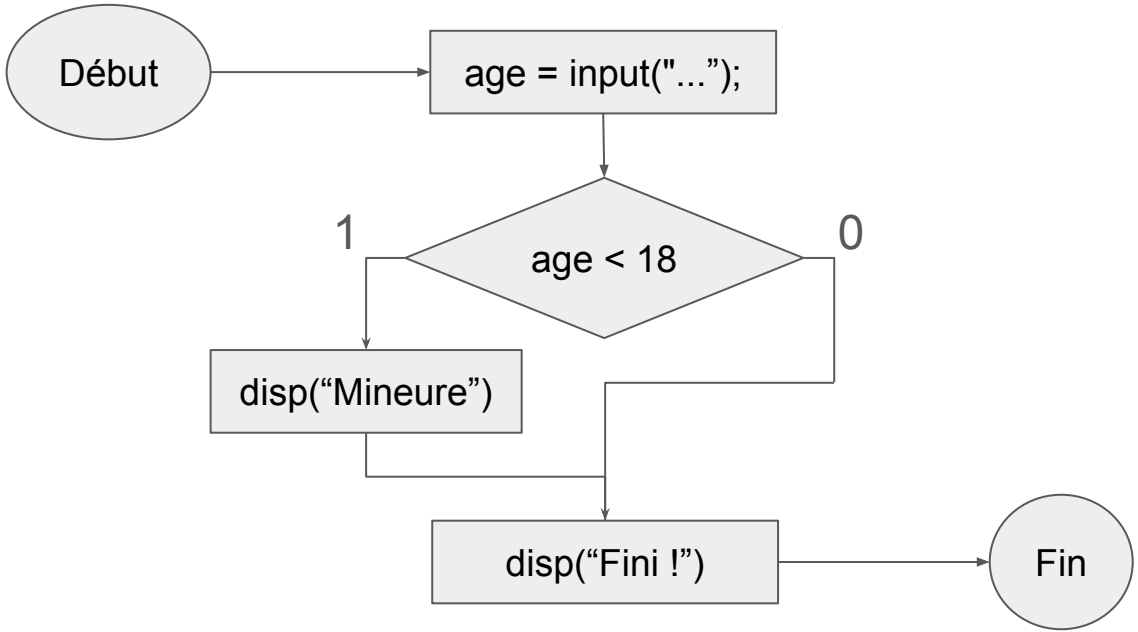
Sélection



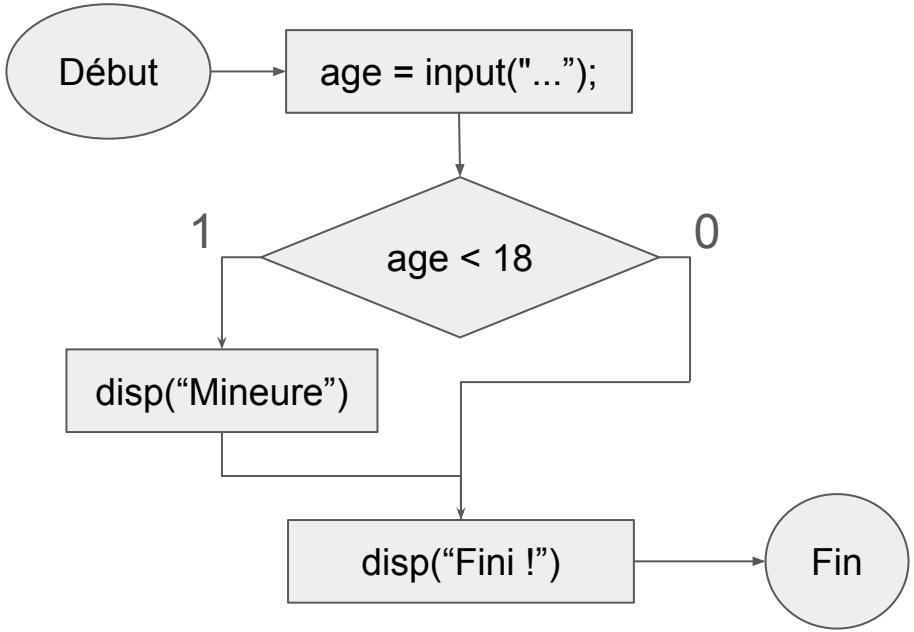
Représentation courante au sein des algorigrammes



Algorigrammes | **Exemple complet avec un *if***



Algorigrammes | **Exemple complet avec un *if***



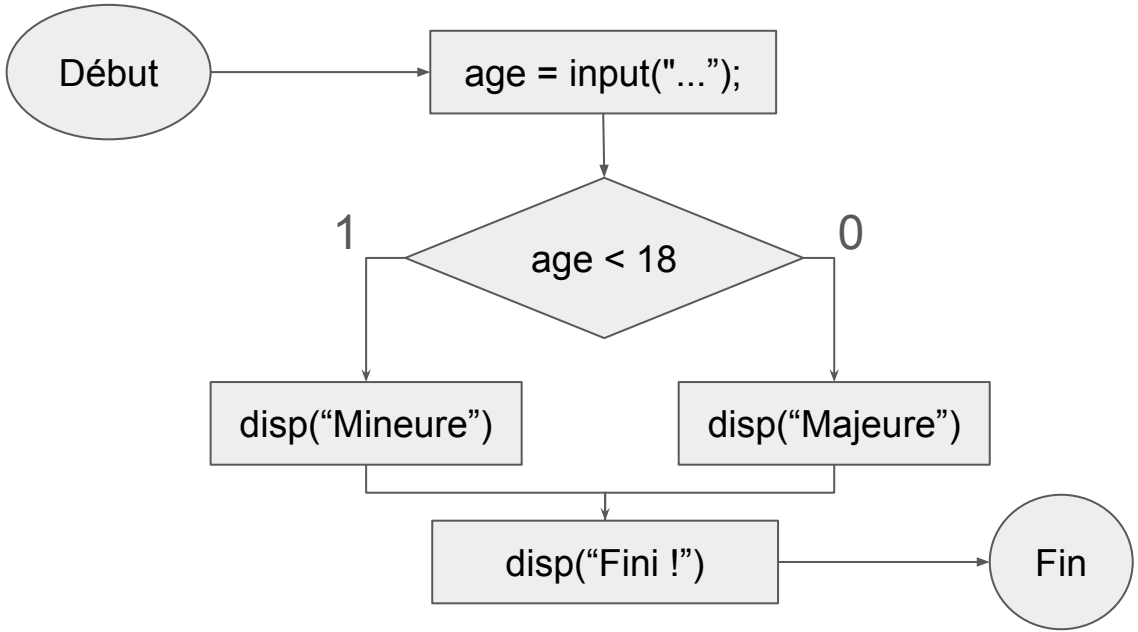

```
age = input("...");
if age < 18
    disp("Mineure")
end
disp(“Fini !”)
```

Le mot-clé *else*

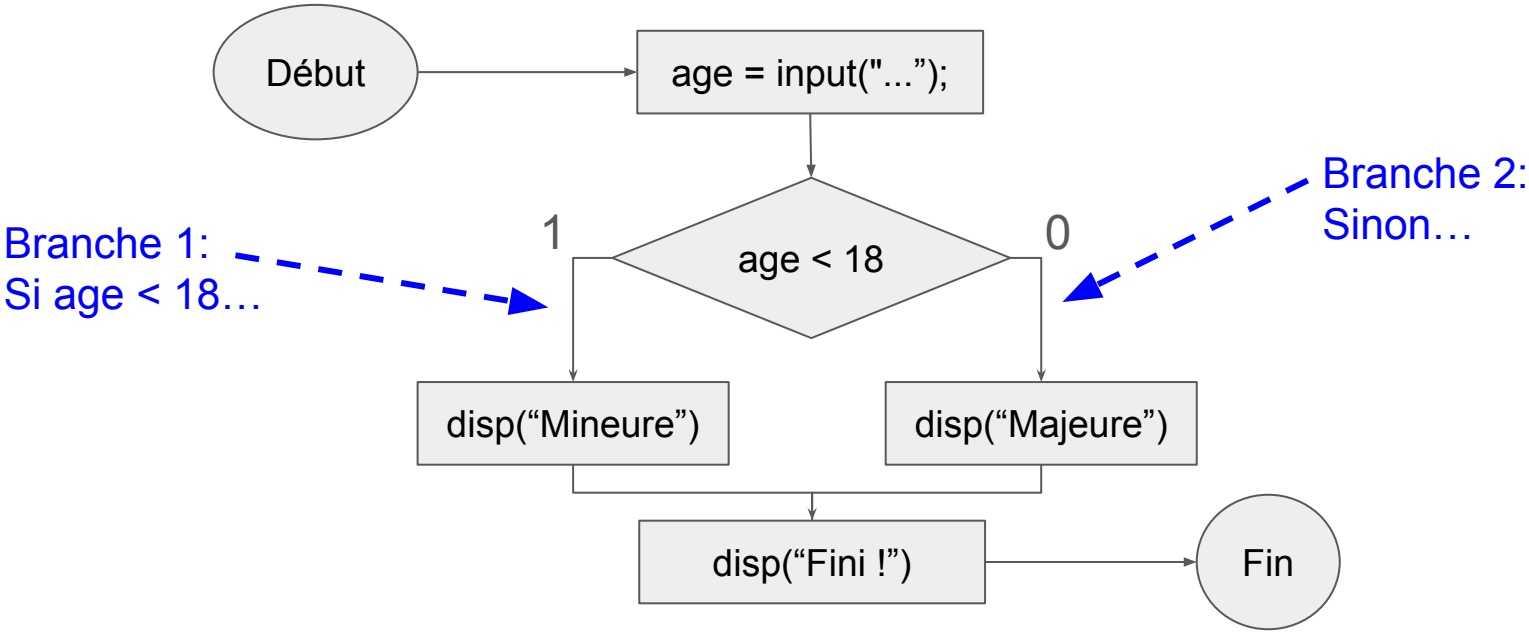
- Équivalent à “sinon”.
- Exemple :

```
age = input("...");
if age < 18
    disp("Mineure")
else
    disp(“Majeure”)
end
disp(“Fini !”)
```

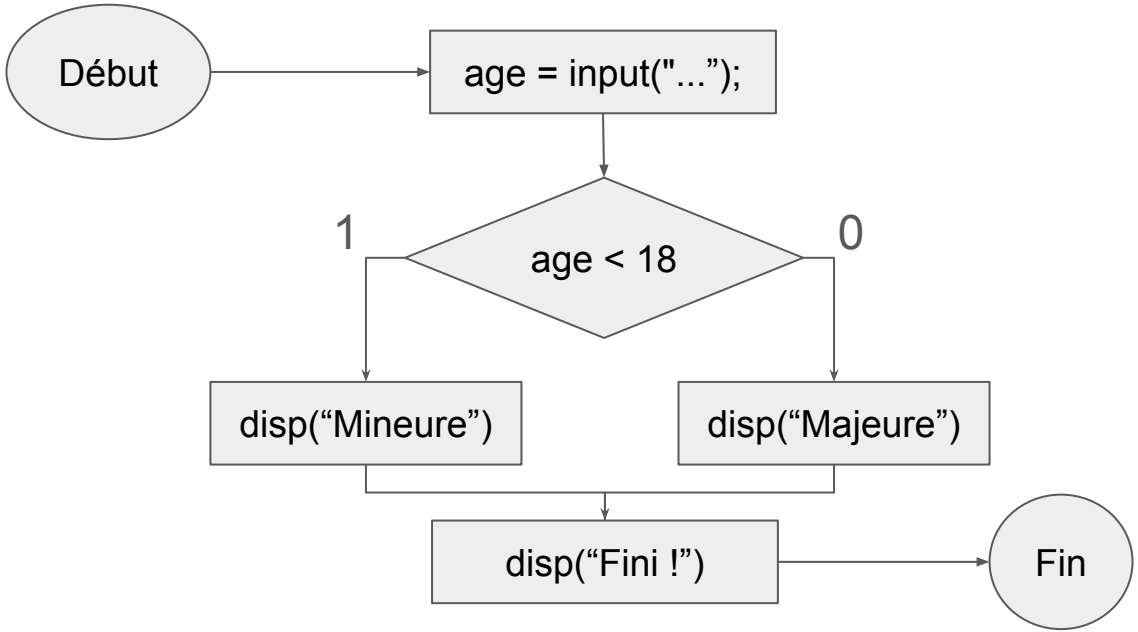
Algorigrammes | **Exemple complet avec *if - else***



Algorigrammes | **Exemple complet avec *if - else***



Algorigrammes | **Exemple complet**




```
age = input("...");
if age < 18
    disp("Mineure")
else
    disp(“Majeure”)
end
disp(“Fini !”)
```

Note sur les blocs

- Indentation des instructions concernées : question de convention de présentation (lisibilité, esthétique).

- Autant d'instructions que l'on veut au sein d'un bloc !

```
age = input("...");
if age < 18
    instruction 1
    instruction 2
    instruction 3
    …
end
```

Blocs imbriqués

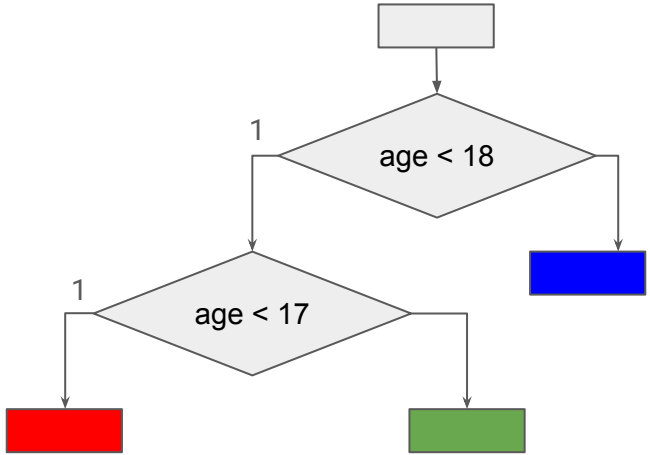
- Permet des sélections multiples. Par exemple :

```
age = input("Indiquez votre âge:");
if age < 18
    if age < 17
          disp("Conduite interdite")
    else
        disp("Conduite accompagnée ok.")
    end
else
    disp("Conduite autorisée.")
end
```

Blocs imbriqués

- Permet des sélections multiples. Par exemple :

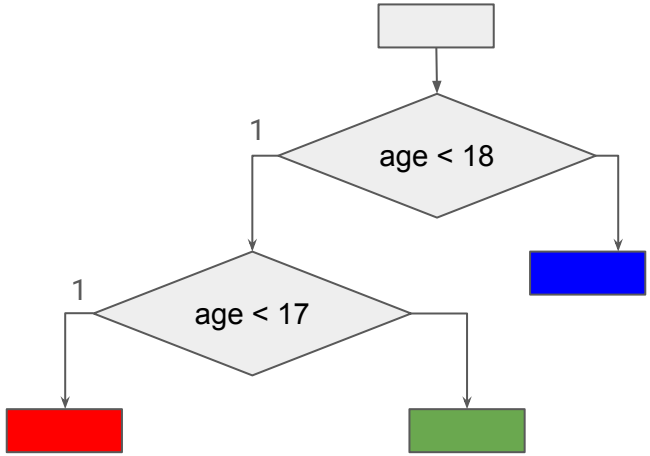
```
age = input("Indiquez votre âge:");
if age < 18
    if age < 17
        disp("Conduite interdite")
    else
        disp("Conduite accompagnée ok.")
    end
else
    disp("Conduite autorisée.")
end
```



Alternative aux blocs imbriqués | **mot-clé *elseif***

- Permet des sélections multiples. Par exemple :

```
age = input("Indiquez votre âge:");
if age < 17
    disp("Conduite interdite")
elseif age < 18
    disp("Conduite accompagnée ok.")
else
    disp("Conduite autorisée.")
end
```



Opérateurs de comparaison

Symbole	Signification
==	égal à
~=	différent de
<	plus petit que
<=	plus petit ou égal à
>	plus grand que
>=	plus grand ou égal à

Opérateurs logiques

Symbole	Signification
`&&`	ET
`||`	OU
`0 ou false`	est faux
`1 ou true`	est vrai

Utilisation des opérateurs logiques

- Permet de combiner des opérateurs de comparaison. Exemple :

```
if x > 10 && x < 20
    disp(“x est entre 10 et 20 (non compris)”)
end
```

- Contrôle de l’ordre des opérations via des parenthèses. Exemple:

```
if x < 7 || (x > 11 && x < 50)
    disp(“x est plus petit que 7 ou alors entre 11 et 50”)
end
```

Variables booléennes

- Le résultat d'une comparaison peut se stocker dans une variable booléenne (deux valeurs possible : soit fausse, soit vraie).

- En Matlab, 0 s'entend comme "faux" ; le reste vaut pour "vrai".
  Exemple à tester:

```
x = -3;
if x
    disp("Vrai")
else
    disp("Faux")
end
```

Variables booléennes

- Permet de rendre le code plus lisible et d'éviter de refaire des calculs.

- Exemple :

```
x = 334637271912712746561 7;
x_est_premier = test_premier(x); % coûteux !
if x > 10 && x_est_premier
    % ...
elseif x > 100 && x_est_premier
    % ...
elseif x > 1000
    % ...
end
```