

# Algorithmique

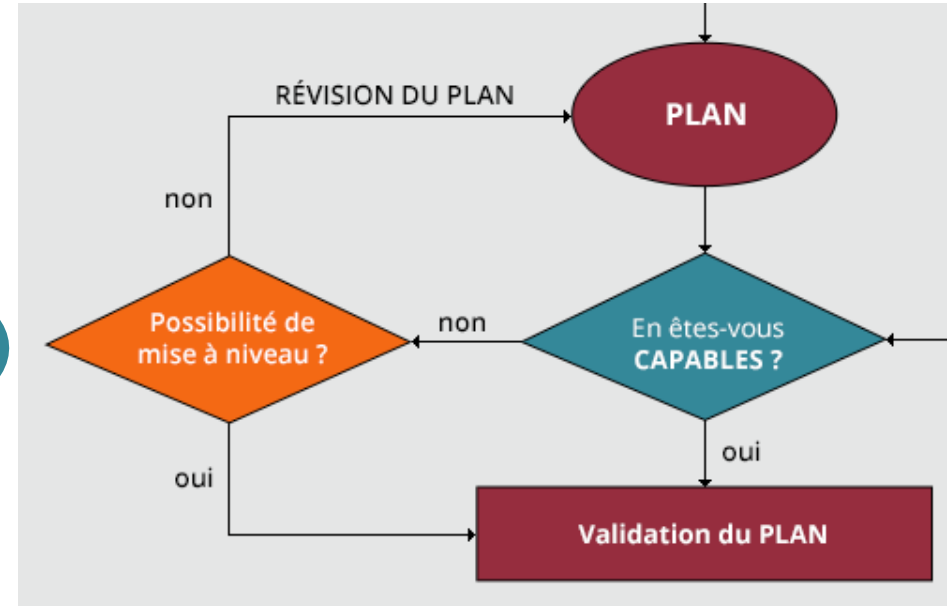
## **Module 4 - Les instructions itératives**

# Objectifs

- Comprendre le principe des instructions itératives
- Savoir utiliser les structures itératives
- Connaître les caractéristiques de chaque boucle et savoir laquelle utiliser en fonction de la situation
- Appréhender les boucles imbriquées

# Les instructions itératives

## Introduction



# La boucle Pour

**Pour** var **<-** ini **à** fin [**par** pas]  
instructions

## **FPour**

- Répète les instructions un certain nombre de fois
  - Exemple :

**Pour** i **<-** 1 **à** 3  
    écrire("Passage n°" **&** i)

## **FPour**



Passage n°1  
Passage n°2  
Passage n°3

# TD : Moyenne de notes

- Affichages et *saisies*



Entrez le nombre de valeurs :

*3*

Valeur :

*12*

Valeur :

*17,5*

Valeur :

*14*

La moyenne est 14,5

Les instructions itératives

# TD : Moyenne de notes

**Algo** moyenneDesNotes

# Demande des valeurs et calcule de la moyenne

Variable nbVal, cpt : entier

Variable moyenne, saisie : réel

**Début**

moyenne <- 0

nbVal <- saisir("Entrer le nombre de notes : ")

**Si** nbVal  $\leq$  0 **Alors**

    écrire("Le nombre de valeurs doit être strictement positif")

**Sinon**

**Pour** cpt <- 1 à nbVal

        saisie <- saisir("Valeur : ")

        moyenne <- moyenne + saisie / nbVal

**FPour**

    écrire("La moyenne est " & moyenne)

**FSi**

**Fin**

Les instructions itératives

# La boucle TantQue

amorçage

**TantQue** condition

instructions

relance

**FTq**

- Répète les instructions tant que la condition vaut **VRAI**
- Les instructions d'amorçage permettent d'initialiser la variable sur laquelle porte la condition
- La relance permet de mettre à jour la variable sur laquelle porte la condition

## Les instructions itératives

# Exemple

#Amorçage

```
saisie <- saisir("Quelle est la capitale de la France ?")
```

#Condition

**TantQue** saisie  $\neq$  "Paris"

#Traitement

```
écrire("Vous vous êtes trompé ! Réessayez...")
```

#Relance

```
saisie <- saisir("Capitale de la France ? ")
```

**FTq**



Quelle est la capitale de la France ? Lyon  
Vous vous êtes trompé ! Réessayez...  
Capitale de la France ? Nantes  
Vous vous êtes trompé ! Réessayez...  
Capitale de la France ? Paris



# TD : Moyenne de notes (version 2)

- Saisir des notes jusqu'à ce que l'utilisateur saisisse -1 et en calculer la moyenne



Note (-1 pour terminer) ?

9

Note (-1 pour terminer) ?

18

Note (-1 pour terminer) ?

-1

La moyenne des notes est de 13,5

# TD : Moyenne de notes (version 2)

**Algo** MoyenneDesNotesV2

#Calcule la moyenne des notes saisies par l'utilisateur

#L'utilisateur saisit -1 pour finir la saisie

Variable nbVal : entier <- 0

Variable saisie, total : réel

Constante STOP : entier <- -1

**Début**

total <- 0

saisie <- saisir("Note (" & STOP & "pour terminer) ?")

**TantQue** saisie ≠ STOP

total <- total + saisie

nbVal <- nbVal + 1

saisie <- saisir("Note (" & STOP & "pour terminer) ?")

**FTq**

**Si** nbVal ≠ 0 **Alors**

écrire("La moyenne des notes est de " & total/nbVal)

**Sinon**

écrire("Aucune note n'a été saisie")

**FSi**

**Fin**

# TD : Moyenne de notes (version 3)

- Modifier votre algorithme pour afficher en plus le pourcentage de notes au-dessus de 10
  - Exemple



Il y a 50 % des notes au-dessus de 10/20.

# TD : Moyenne de notes (version 3)

**Algo** MoyenneDesNotesV3      #Calcule la moyenne et donne le pourcentage de note  $\geq 10$

**Variable** nbVal  $\leftarrow 0$ , nbS  $\leftarrow 0$  : entier

**Variable** saisie, total  $\leftarrow 0$  : réel

**Constante** STOP : entier  $\leftarrow -1$

**Constante** PALIER : entier  $\leftarrow 10$

**Début**

saisie  $\leftarrow$  saisir("Note (" & STOP & "pour terminer) ?")

**TantQue** saisie  $\neq$  STOP

total  $\leftarrow$  total + saisie

nbVal  $\leftarrow$  nbVal + 1

**Si** saisie  $\geq$  PALIER **Alors**

nbS  $\leftarrow$  nbS + 1

**FSi**

saisie  $\leftarrow$  saisir("Note ? ")

**FTq**

**Si** nbVal  $\neq 0$  **Alors**

écrire("Moyenne = " & total/nbVal & " (" & nbS/nbVal  $\times$  100 & "%  $\geq$  " & PALIER & ")")

**Sinon**

écrire("Aucune note n'a été saisie")

**FSi**

**Fin**

# TD : Devinez à quel nombre je pense

- Affichages et *saisies*



À quel nombre entre 1 et 100 je pense ?

*23*

C'est moins !

*15*

C'est plus !

*19*

C'est moins !

*17*

Bravo ! Vous avez trouvé !

Les instructions itératives

# TD : Devinez à quel nombre je pense

**Algo** nombreCacheCache

Constante NB\_MYSTERE : entier <- aléa(1,100) #initialisation aléatoire

Les instructions itératives

# TD : Devinez à quel nombre je pense...

**Algo** nombreCacheCache

**Constante** NB\_MYSTERE : entier <- aléa(1,100) #initialisation aléatoire

**Variable** saisie : entier

**Début**

saisie <- saisir("A quel nombre entre 1 et 100 je pense ? ")

**TantQue** saisie ≠ NB\_MYSTERE

**Si** saisie > NB\_MYSTERE **Alors**

        écrire("C'est moins !")

**Sinon**

        écrire("C'est plus !")

**FSi**

        saisie <- saisir()

**FTq**

écrire("Bravo ! Vous avez trouvé !")

**Fin**

# TD : Que fait cet algorithme ?

**Algo** Piege

Variable nbTentatives : entier <- 1

Variable saisie : texte

Constante MAX\_TENTATIVES : entier <- 5

**Début**

saisie <- saisir("Quelle est la capitale de la France ? ")

**TantQue** saisie ≠ "Paris" **ou** MAX\_TENTATIVES - nbTentatives ≠ 0

    écrire("Mauvaise réponse !")

    écrire("Plus que " & MAX\_TENTATIVES - nbTentatives & " tentative(s)")

    saisie <- saisir("Quelle est la capitale de la France ? ")

**FTq**

**Si** MAX\_TENTATIVES - nbTentatives ≠ 0 **Alors**

    écrire("Bravo !")

**Sinon**

    écrire("Revoyez votre géographie !")

**FSi**

**Fin**



# TD : Que fait cet algorithme ?

## Une première erreur corrigée

**Algo** UneErreurDeMoins

**Variable** nbTentatives : entier <- 1

**Variable** saisie : texte

**Constante** MAX\_TENTATIVES : entier <- 5

**Début**

saisie <- saisir("Quelle est la capitale de la France ? ")

**TantQue** saisie ≠ "Paris" **ou** MAX\_TENTATIVES - nbTentatives ≠ 0

    écrire("Mauvaise réponse !")

    écrire("Plus que " & MAX\_TENTATIVES - nbTentatives & " tentative(s)")

    saisie <- saisir("Quelle est la capitale de la France ? ")

    nbTentatives <- nbTentatives + 1

**FTq**

**Si** MAX\_TENTATIVES - nbTentatives ≠ 0 **Alors**

    écrire("Bravo !")

**Sinon**

    écrire("Revoyez votre géographie !")

**FSi**

**Fin**

# TD : Que fait cet algorithme ?

## Une seconde erreur corrigée

**Algo** DeuxErreursDeMoins

**Variable** nbTentatives : entier <- 1

**Variable** saisie : texte

**Constante** MAX\_TENTATIVES : entier <- 5

**Début**

saisie <- saisir("Quelle est la capitale de la France ? ")

**TantQue** saisie ≠ "Paris" **et** MAX\_TENTATIVES - nbTentatives ≠ 0

    écrire("Mauvaise réponse !")

    écrire("Plus que " & MAX\_TENTATIVES - nbTentatives & " tentative(s)")

    saisie <- saisir("Quelle est la capitale de la France ? ")

    nbTentatives <- nbTentatives + 1

**FTq**

**Si** MAX\_TENTATIVES - nbTentatives ≠ 0 **Alors**

    écrire("Bravo !")

**Sinon**

    écrire("Revoyez votre géographie !")

**FSi**

**Fin**

# TD : Que fait cet algorithme ?

## Une troisième erreur corrigée

**Algo** TroisErreursDeMoins

Variable nbTentatives : entier <- 1

Variable saisie : texte

Constante MAX\_TENTATIVES : entier <- 5

**Début**

saisie <- saisir("Quelle est la capitale de la France ? ")

**TantQue** saisie ≠ "Paris" **et** MAX\_TENTATIVES - nbTentatives ≠ 0

    écrire("Mauvaise réponse !")

    écrire("Plus que " & MAX\_TENTATIVES - nbTentatives & " tentative(s)")

    saisie <- saisir("Quelle est la capitale de la France ? ")

    nbTentatives <- nbTentatives + 1

**FTq**

**Si** saisie = "Paris" **Alors**

    écrire("Bravo !")

**Sinon**

    écrire("Revoyez votre géographie !")

**FSi**

**Fin**

# La boucle Répéter

## Répéter

(ré)affectation de la variable de condition  
instructions

**TantQue** condition **FRépéter**

- Exécute au moins une fois la boucle
  - Exemple :

## Répéter

```
nb <- saisir("Un nombre pair, svp")
```

**TantQue** nb **mod** 2 **≠** 0 **FRépéter**

Les instructions itératives

# TD : Affichage de répliques de films

- Le programme affiche un menu avec une liste de films et une option pour quitter
  - Si l'utilisateur choisit l'un des films
    - le programme affiche une réplique de ce film
    - l'utilisateur peut choisir un autre film ou quitter
  - Si l'utilisateur choisit de quitter
    - Le programme affiche un message avant de se terminer

# TD : Affichage de répliques de films

- Affichages et *saisies*



1 – Une réplique de la cité de la peur

2 – Une réplique de James Bond

3 – Une réplique de la vie est un long fleuve tranquille

4 – Une réplique de Star Wars

5 – Quitter cette magnifique application !

2

Je m'appelle Bond, James Bond

1

Attention, c'est une véritable boucherie !

5

Au revoir !

# TD : Affichage de répliques de films

**Algo** repliquesDeFilms

Variable saisie : entier

**Début**

écrire("1–Une réplique de la cité de la peur")

écrire("2–Une réplique de James Bond")

écrire("3–Une réplique de la vie est un long fleuve tranquille")

écrire("4–Une réplique de Star Wars")

écrire("5–Quitter cette magnifique application !")

**Répéter**

saisie <- saisir()

**Selon** saisie

**cas** 1 : écrire("Attention, c'est une véritable boucherie !")

**cas** 2 : écrire("Je m'appelle Bond, James Bond !")

**cas** 3 : écrire("Le lundi, c'est ravioli")

**cas** 4 : écrire("La force soit avec toi, jeune padawan !")

**cas** 5 : écrire("Au revoir !")

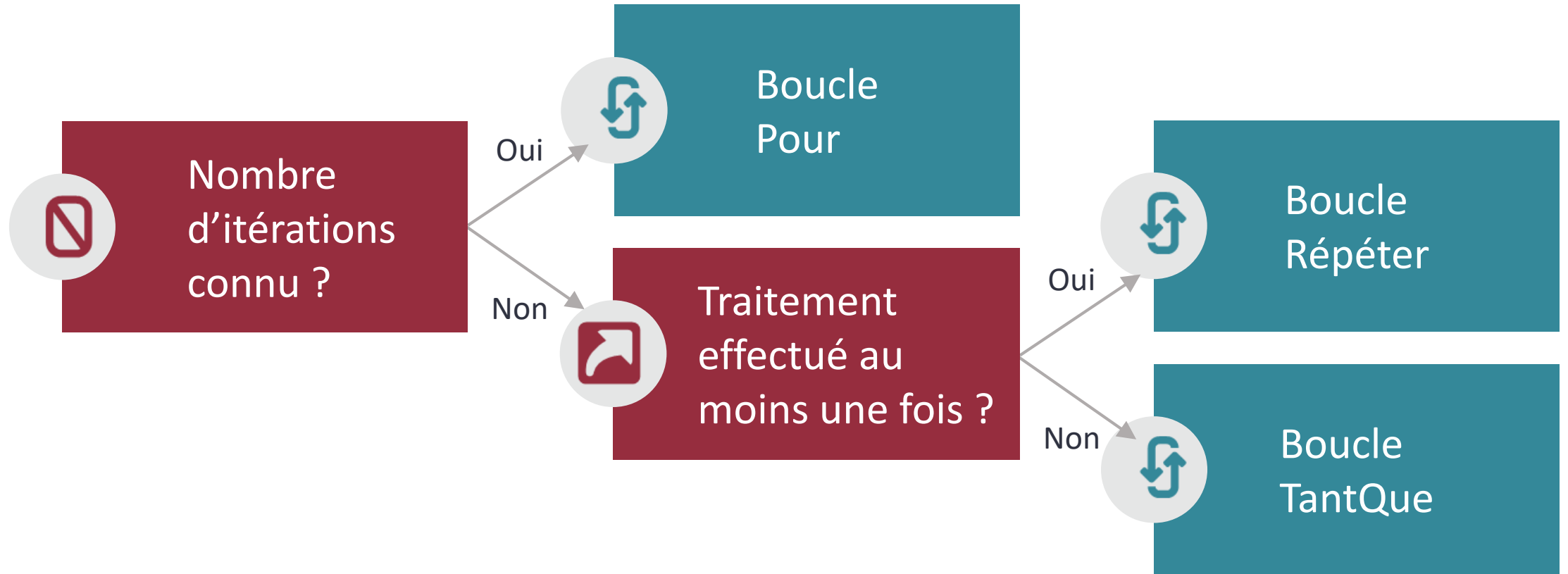
**autre** : écrire("Saisie incorrecte")

**FSelon**

**TantQue** saisie ≠ 5 **FRépéter**

**Fin**

# Quelle boucle choisir ?





# TD : À moi de trouver

- L'ordinateur doit deviner un nombre choisi par l'utilisateur
  - L'utilisateur lui indique si le nombre est plus grand (+), plus petit (-) ou s'il a trouvé le nombre (=)
  - Affichages et *saisies*



Choisissez un nombre compris entre 1 et 100, puis appuyez sur une touche

*a*

Je tente 45, est-ce plus, moins ou est-ce le nombre (+/-/=) ?

-

Je tente 10, est-ce plus, moins ou est-ce le nombre (+/-/=) ?

+

Je tente 22, est-ce plus, moins ou est-ce le nombre (+/-/=) ?

+

Je tente 27, est-ce plus, moins ou est-ce le nombre (+/-/=) ?

=

Super ! J'ai trouvé en 4 tentatives

## Les instructions itératives

# TD : À moi de trouver

**Algo** devineUnNombre

Variable min : entier <- 1

Variable max : entier <- 100

Variable reponse : caractère

Variable nbTentatives : entier <- 0

Variable valeur : entier

Variable saisieOk : booléen

**Début** # le caractère saisi n'a pas d'importance. C'est pour attendre que l'utilisateur ai choisi son nombre  
saisir("Choisissez un nombre compris entre " & min & " et " & max & ", puis appuyez sur une touche")

**Répéter**

valeur <- aléa(min, max)

écrire("Je tente " & valeur & ", est-ce plus, moins ou est-ce le nombre (+/-/=) ?")

nbTentatives <- nbTentatives + 1

**Répéter**

saisieOk <- VRAI

reponse <- saisir()

**Selon** reponse

cas '+' : min <- valeur

cas '-' : max <- valeur

cas '=' : écrire("Super ! J'ai réussi en " & nbTentatives & " tentatives")

autre : saisieOk <- FAUX

écrire("Erreur de saisie. Saisissez + - ou =")

**FSelon**

**TantQue non** saisieOk **FRépéter**

**TantQue** reponse ≠ '=' **FRépéter**

**Fin**

# TD : Saisie d'un multiple de 3

- Affichages et *saisies*



Entrez un multiple de 3

*7579*

Erreur 7579 n'est pas un multiple de 3

*16427*

Erreur 16427 n'est pas un multiple de 3

*51321*

Ok : 51321 est un multiple de 3

Les instructions itératives

# TD : Saisie d'un multiple de 3

**Algo** multiple

Variable saisie : entier

Constante MULTIPLE : entier <- 3

**Début**

    écrire("Entrer un multiple de " & MULTIPLE)

    saisie <- saisir()

**TantQue** saisie mod MULTIPLE ≠ 0

        écrire("Erreur " & saisie & " n'est pas un multiple de " & MULTIPLE)

        saisie <- saisir()

**FTq**

    écrire("Ok " & saisie & " est un multiple de " & MULTIPLE)

**Fin**

# TD : Rendez la monnaie !

- Programmer le monnayeur de la machine à café
  - Prix du café : 0,60 €
  - Pièces acceptées : 2 € 1 € 0,50 € 0,20 € 0,10 € 0,05 €
  - Affichages et *saisies*



Entrez la valeur de la pièce :

*2*

Voici votre café et votre monnaie (1,40€) :

1 pièce(s) de 1€

2 pièce(s) de 0,20€

- Autres affichages et *saisies* possibles



Entrez la valeur de la pièce :

*0,20*

Crédit insuffisant (0,20€/0,60€)

*0,01*

Pièce non acceptée, entrez une autre pièce

*0,20*

Crédit insuffisant (0,40€/0,60€)

*0,50*

Voici votre café et votre monnaie (0,30 €) :

1 pièce(s) de 0,20€

1 pièce(s) de 0,10€

# TD : Rendez la monnaie !

**Algo** monnaieCafe

# Récupère des pièces jusqu'à ce que le crédit soit suffisant puis rend la monnaie

Variable credit : réel <- 0

Variable nbPieces : entier

Variable saisie : réel

Constante PRIX\_CAFE : réel <- 0,60

**Début**

écrire("Entrez la valeur de la pièce : ")

**Répéter**

saisie <- saisir()

# Si la pièce est acceptable

**Selon**

**cas** 2 ; 1 ; 0,5 ; 0,2 ; 0,1 ; 0,05 :

credit <- credit + saisie

**Si** credit < PRIX\_CAFE **Alors**

écrire("Crédit insuffisant (" & credit & "€/ " & PRIX\_CAFE & "€)")

**FSi**

**autre :**

écrire("Pièce non acceptée, entrez une autre pièce")

**FSelon**

**TantQue** credit < PRIX\_CAFE **FRépéter**

credit <- credit - PRIX\_CAFE

écrire("Voici votre café et votre monnaie (" & credit & "€) :")

# TD : Rendez la monnaie !

```
# Rendu de la monnaie en pièces de 1€
nbPieces <- 0
TantQue credit ≥ 1
    nbPieces <- nbPieces + 1
    credit <- credit - 1
FTq
Si nbPieces > 0 Alors
    écrire(nbPieces & "pièce(s) de 1€")
FSi

# Rendu de la monnaie en pièces de 0,50€
nbPieces <- 0
TantQue credit ≥ 0,5
    nbPieces <- nbPieces + 1
    credit <- credit - 0,5
FTq
Si nbPieces > 0 Alors
    écrire(nbPieces & "pièce(s) de 0,50€")
FSi
```

# TD : Rendez la monnaie !

```
# Rendu de la monnaie en pièces de 0,20€
nbPieces <- 0
TantQue credit ≥ 0,2
    nbPieces <- nbPieces + 1
    credit <- credit - 0,2
FTq
Si nbPieces > 0 Alors
    écrire(nbPieces & "pièce(s) de 0,20€")
FSi

# Rendu de la monnaie en pièces de 0,10€
nbPieces <- 0
TantQue credit ≥ 0,1
    nbPieces <- nbPieces + 1
    credit <- credit - 0,1
FTq
Si nbPieces > 0 Alors
    écrire(nbPieces & "pièce(s) de 0,10€")
FSi
```



Les instructions itératives

# TD : Rendez la monnaie !

```
# Rendu de la monnaie en pièces de 0,05€
```

```
nbPieces <- 0
```

```
TantQue credit ≥ 0,05
```

```
    nbPieces <- nbPieces + 1
```

```
    credit <- credit - 0,05
```

```
FTq
```

```
Si nbPieces > 0 Alors
```

```
    écrire(nbPieces & "pièce(s) de 0,05€")
```

```
FSi
```

```
Fin
```

# TD : ASCII Art !

- Dessiner un rectangle



Largeur ?

8

Hauteur?

5

Caractère?

#

#####

#####

#####

#####

#####



écrireSRC()

permet d'écrire sans  
retourner à la ligne

## Les instructions itératives

# TD : ASCII Art !

**Algo** dessin

# Affiche un rectangle rempli avec un même caractère

Variable largeur, hauteur, i, j : entier

Variable car : caractère

**Début**

largeur <- saisir("Largeur ? ")

hauteur <- saisir("Hauteur ? ")

car <- saisir("Caractère ? ")

**Pour** j <- 1 à hauteur

**Pour** i <- 1 à largeur

écrireSRC(car)

**FPour**

écrire()

**FPour**

**Fin**



Largeur ?

8

Hauteur?

5

Caractère?

#

#####

#####

#####

#####

#####

# TD : Formes géométriques

- Dessiner une des formes suivantes selon le choix de l'utilisateur



taille ?

5

Caractère ?

#

forme ?

1 – rectangle plein

2 – rectangle creux

3 – croix de Saint-André

4 – triangle

5 – losange

6 – damier

2

```
#####
#      #
#      #
#      #
#####
```

```
#####
#####
#####
#####
#####
```

Rectangle  
plein

```
#
##
# #
#  #
####
```

Triangle

```
#  #
#  #
#
#  #
#  #
```

Croix de  
Saint-André

```
#
# #
#  #
#  #
#
```

Losange

```
# # #
#  #
#  # #
#  #
#  # #
```

Damier



Remarque : le nombre  
de lignes est égal au  
nombre de colonnes

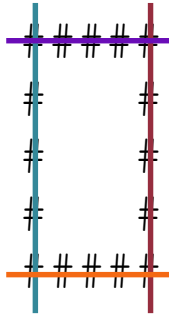
# TD : Formes géométriques

Numéro de ligne = 1

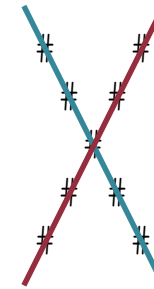
Numéro de ligne  
= Taille

Numéro de colonne = 1

Numéro de colonne  
= Taille



Numéro de colonne = Numéro de ligne



Numéro de colonne + Numéro de ligne  
= Taille + 1

# TD : Rendez la monnaie (version 2)

- Rendez la monnaie (version 2)

- Gérer un stock de pièces

- Initialement vous avez un stock de pièces

- Par exemple

2 €	1 €	0€50	0€20	0€10	0€05
10	3	0	10	10	5

- Modifier ce stock en fonction des pièces insérées et des pièces distribuées