

Didactique de l'informatique

M1 MEEF NSI

La variable en informatique

Benjamin Wack



2024

Vos conceptions

Quelles sont, selon vous, les difficultés posées par la notion de variable en informatique ?

Plan

Difficultés et Compétences des élèves

Comment s'approprier la notion de variable

Une piste pour expliquer certaines difficultés

Une activité (pour des élèves de collège) : Retour au port (*Zebiche & Declercq, 2019*)

<https://scratch.mit.edu/projects/315299642>

Consigne élève :

- ▶ On vous donne un paysage, des blocs de programme pour manipuler un bateau.
- ▶ Défi : chercher le plus court chemin possible pour vous rendre au port
- ▶ Modifier le programme pour mesurer automatiquement le chemin parcouru
- ▶ Modifier le programme pour que le bateau n'avance plus s'il heurte un rocher

Analyse *a priori*

- ▶ Quels sont les objectifs de cette activité ?
- ▶ Quels sont les prérequis ?
- ▶ Que remarquez-vous à propos de la consigne ?

Analyse *a priori*

- ▶ Quels sont les objectifs de cette activité ?
- ▶ Quels sont les prérequis ?
- ▶ Que remarquez-vous à propos de la consigne ?

- ▶ Proposer (et tester) une solution
- ▶ Quelles sont les stratégies possibles pour les élèves ?
- ▶ À quelles difficultés s'attendre ? Quelles aides apporter ?

Analyse *a priori*

- ▶ Quels sont les objectifs de cette activité ?
- ▶ Quels sont les prérequis ?
- ▶ Que remarquez-vous à propos de la consigne ?

- ▶ Proposer (et tester) une solution
- ▶ Quelles sont les stratégies possibles pour les élèves ?
- ▶ À quelles difficultés s'attendre ? Quelles aides apporter ?

- ▶ Quelle mise en œuvre dans la classe ?

Productions d'élèves

- ▶ Identifier les erreurs
- ▶ Interpréter : qu'a voulu faire l'élève ?
quelle conception erronée cette erreur révèle-t-elle ?

Et après

- ▶ Quelle institutionnalisation à la suite de cette séance ?
- ▶ Quelle place dans la progression en algo/prog ?

Compétences des élèves (*Samurçay, 1985*)

Expérience réalisée sur des élèves de seconde de niveaux variés après 15h d'introduction à la programmation : variables entières, réelles, tableaux ; boucles REPEAT UNTIL et FOR ; lecture et écriture.

Consigne

Le programme suivant calcule $x^n - y^n$ pour x, y, n entiers positifs.

```
var x, y, n, expox, expoy, compteur, resultat: integer;
```

```
read(x); read(y); read(n);
```

```
#
```

```
repeat
```

```
    expox := expox * x ;
```

```
    expoy := expoy * y ;
```

```
    compteur := compteur + 1;
```

```
until compteur = n;
```

```
resultat := expox - expoy;
```

```
write ('le resultat est: ', resultat);
```

Complétez si nécessaire la ou les lignes manquantes là où est le signe #.

Réponses des élèves

initialisent correctement les trois variables	1
répondent "non" il ne manque pas d'instruction	4
répondent "oui" mais n'arrivent pas à identifier les variables	3
initialisent partiellement	12
- COMPTEUR seulement	7
- EXPOX seulement	1
- EXPOX et COMPTEUR seulement	1
- RESULTAT seulement	3
initialisation erronée	4
- read (EXPOX); read(EXPOY)	2
- read (EXPOX)	1
- n := 0	1
sans réponse	2

Que démontre cette étude ?

Analyse des résultats

Pour catégoriser, Samurçay distingue différents types de variables :

- ▶ les variables correspondant à une donnée explicite du problème
- ▶ les variables qui reproduisent un traitement manuel
- ▶ les variables purement internes, propres au traitement informatique

et les décisions à prendre pour les manipuler :

- ▶ initialisation
- ▶ mise à jour (particulièrement spécifique de la variable informatique)
- ▶ test (d'arrêt)

Analyse des résultats

Pour catégoriser, Samurçay distingue différents types de variables :

- ▶ les variables correspondant à une donnée explicite du problème
- ▶ les variables qui reproduisent un traitement manuel
- ▶ les variables purement internes, propres au traitement informatique

et les décisions à prendre pour les manipuler :

- ▶ initialisation
- ▶ mise à jour (particulièrement spécifique de la variable informatique)
- ▶ test (d'arrêt)

Points marquants

- ▶ Réussites : compteurs simples, initialisation des *variables données*
- ▶ Difficultés : *variables internes*, autres initialisations
- ▶ Trois conceptions erronées :
 - ▶ « on doit dire à la machine les valeurs des variables »
 - ▶ « il faut vider les cases mémoire au début de chaque programme »
 - ▶ « un compteur s'initialise à 0 et s'incrémente de 1 »

Plan

Difficultés et Compétences des élèves

Comment s'approprier la notion de variable

Une piste pour expliquer certaines difficultés

Généraliser l'institutionnalisation

Une piste intéressante : les compteurs sont mieux traités car institutionnalisés (par exemple lors de l'enseignement de la boucle). Pour aller plus loin, on peut distinguer plusieurs rôles pour les variables.

Limites du cadre proposé

- ▶ On caractérise ici le *comportement* (valeurs successives) des variables et pas leur *utilisation* (opérande d'un calcul, borne d'une boucle...).
- ▶ D'autres rôles peuvent être identifiés lorsqu'on manipule des tableaux ou structures de données.

Exemple : Occurrences d'une valeur dans un tableau

```
const last = 7;

var value: ArrayType; (* le tableau *)
    i, (* indice *)
    key, (* valeur recherchée *)
    count: integer; (* nombre d'occurrences *)

writeln('Entrez ', last, ' valeurs :');
for i := 1 to last do read(value[i]);

write('Entrez la valeur à chercher :'); readln(key);

count := 0;
for i := 1 to last do
    if value[i] = key then count := count + 1;

writeln('Il y a ', count, ' occurrences de ',
        key, ' dans le tableau');
```


Exemple : Tracé d'histogramme

```
const longest = 40; (* Taille max d'une barre *)

var amount : array [1..12] of real; (* Données à tracer *)
    max : real; (* Valeur maximale *)
    month, (* Mois courant *)
    i : integer;

for month := 1 to 12 do begin
    write('Entrez le montant pour le mois ', month );
    readln(amount[month]);
    if i = 1 then max := amount[1]
        else if max < amount[month] then max := amount[month];
end;

for month := 1 to 12 do begin
    write(month);
    for i := 1 to round(amount[month] / max * longest)
        do write('*');
end
```

Exemple : Simulation d'un jeu de dés

```
var die1, die2, (* Valeurs des dés *)
    total1, total2: integer; (* Score de chaque joueur *)
    firstPlayer: boolean; (* À qui le tour ? *)

total1 := 0; total2 := 0;
firstPlayer := True;

while (total1 < 100) and (total2 < 100) do begin
    die1 := Random(6) + 1;
    die2 := Random(6) + 1;
    if firstPlayer then write('Premier') else write('Second');
    writeln(' joueur lance : ', die1, die2);
    if firstPlayer then total1 := total1 + die1 + die2
        else total2 := total2 + die1 + die2;
    firstPlayer := not firstPlayer;
end;

if total1 > total2 then write('Premier') else write('Second');
write(' joueur gagne.');
```

Exemple : Croissance d'un capital placé en banque

```

var capital,          (* Capital sur le compte          *)
    percent, factor, (* Taux d'intérêt et Facteur de croissance *)
    interest: real;   (* Intérêts pour l'année en cours *)
years (* Durée *), i : integer;
inputOk : Boolean;   (* Les données ont-elles un sens *)

write('Capital initial (positif ou négatif) : '); readln(capital);
inputOk := false;
while not inputOk do begin
    write('Taux d'intérêt (%): '); readln(percent);
    write('Durée (années) : '); readln(years);
    inputOk := (percent > 0) and (years > 0);
    if not inputOk then writeln('Données invalides.');
```

end;

```

factor := percent / 100;
for i := 1 to years do begin
    interest := capital * factor;
    capital := capital + interest;
    writeln('Après ', i, 'années, les intérêts sont ',
            interest, ' et le capital est ', capital);
end
```

Exemple : Meilleure moyenne lissée sur 3 mois

```
var month: integer;                (* Mois courant                *)
    current, previous, preceding, (* Données sur 3 mois        *)
    average,                       (* Moyenne courante          *)
    largest: real;                 (* Meilleure moyenne courante *)

write('Entrez valeur 1'); readln(preceding);
write('Entrez valeur 2'); readln(previous);
write('Entrez valeur 3'); readln(current);
largest := (current + previous + preceding) / 3;

for month := 4 to 12 do begin
    preceding := previous;
    previous := current;
    write('Entrez valeur ', month); readln(current);
    average := (current + previous + preceding) / 3;
    if average > largest then largest := average;
end;

writeln('Meilleure moyenne lissée : ', largest);
```

Rôles des variables, d'après (*Sajaniemi, 2005*)

- ▶ Constante
 - ▶ Paramètre exogène, souvent une donnée du problème
 - ▶ ou Paramètre endogène, calculé à partir des données
- ▶ Information la plus Récente
- ▶ Information la plus Pertinente
- ▶ Compteur (de boucle)
- ▶ Accumulateur
- ▶ Temporaire (calculée)
- ▶ Suiveur (recopie d'une autre variable)
- ▶ Drapeau

Avertissement

Il ne s'agit pas de faire apprendre du vocabulaire arbitrairement, mais de donner une grille de lecture pour faciliter l'activité de programmation.

Quelques particularités de Scratch

- ▶ Certaines variables sont prédéfinies **abscisse** **taille** **chronomètre**
- ▶ **réponse** : Information la plus Récente
- ▶ Les instructions prédéfinies sur les variables :

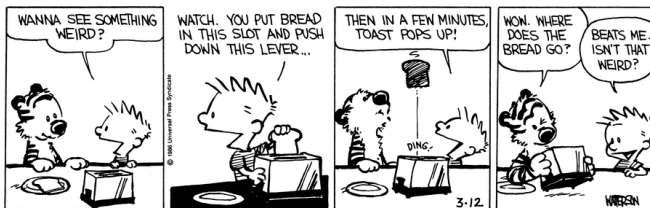
mettre **ma variable** à **0**

ajouter **1** à **ma variable**

incitent à certains types d'utilisations

Notion d'état

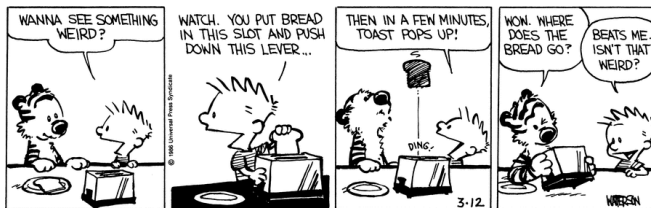
Un programme ne « fait » rien, il « agit sur » l'état d'un système.



Agir sur soit ici être pris au sens large : affectation ; affichage de texte ; déplacement de lutin ; actionneur de robot...

Notion d'état

Un programme ne « fait » rien, il « agit sur » l'état d'un système.



Agir sur soit ici être pris au sens large : affectation ; affichage de texte ; déplacement de lutin ; actionneur de robot...

Difficulté : l'état est caché, mais pour bien programmer il faut se le représenter.

Des réponses possibles :

- ▶ Tortues et assimilés
- ▶ Visualiseurs de variables (simple débogueur ou plus avancé)

Plan

Difficultés et Compétences des élèves

Comment s'approprier la notion de variable

Une piste pour expliquer certaines difficultés

Une difficulté supplémentaire : la variable (et plus généralement la lettre) en mathématiques

Par ordre d'apparition dans le cursus de l'élève :

- ▶ Étiquette (longueur, largeur... dans une formule de périmètre ou d'aire)
- ▶ Indéterminée dans une égalité remarquable (peut prendre toutes les valeurs)
- ▶ Inconnue d'une équation
- ▶ Variable d'une fonction
- ▶ Paramètre d'un problème

Expérience de Briant

Consigne

Écrire un algorithme résolvant l'équation $ax + b = c$

Expérience de Briant

Consigne

Écrire un algorithme résolvant l'équation $ax + b = c$

Quelques productions d'élèves :

► $x = (c-b) / a$

Expérience de Briant

Consigne

Écrire un algorithme résolvant l'équation $ax + b = c$

Quelques productions d'élèves :

- ▶ $x = (c-b) / a$
procédure correcte (sans les cas particuliers : pourquoi?)
- ▶ $I = c$
 $I = I-b$
 $I = I/a$
 $x = I$

Expérience de Briant

Consigne

Écrire un algorithme résolvant l'équation $ax + b = c$

Quelques productions d'élèves :

- ▶ $x = (c-b) / a$
procédure correcte (sans les cas particuliers : pourquoi?)
- ▶ $I = c$
 $I = I-b$
 $I = I/a$
 $x = I$
procédure pas à pas (correcte mais mime la résolution à la main)
- ▶ $c = ax+b$

Expérience de Briant

Consigne

Écrire un algorithme résolvant l'équation $ax + b = c$

Quelques productions d'élèves :

- ▶ $x = (c-b) / a$
procédure correcte (sans les cas particuliers : pourquoi ?)
- ▶ $I = c$
 $I = I-b$
 $I = I/a$
 $x = I$
procédure pas à pas (correcte mais mime la résolution à la main)
- ▶ $c = ax+b$
confusion entre l'affectation et l'équation
- ▶ Si $ax+b=c$
alors Afficher x

Expérience de Briant

Consigne

Écrire un algorithme résolvant l'équation $ax + b = c$

Quelques productions d'élèves :

- ▶ $x = (c-b) / a$
procédure correcte (sans les cas particuliers : pourquoi ?)
- ▶ $I = c$
 $I = I-b$
 $I = I/a$
 $x = I$
procédure pas à pas (correcte mais mime la résolution à la main)
- ▶ $c = ax+b$
confusion entre l'affectation et l'équation
- ▶ Si $ax+b=c$
alors Afficher x
conception de l'ordinateur comme un outil de calcul formel