

Introduction à l'informatique, premiers programmes

A. C'est quoi l'informatique	7
B. À propos de ce cours	39
C. Ordinateurs et traitement automatique des informations	51
D. Premiers programmes	65
E. Expressions	76
F. Variables	84

Pourquoi enseigner l'informatique ?

Pourquoi enseigner l'informatique ?

Évidence : l'ordinateur est partout !

- Combien d'ordinateurs dans la salle ?

Pourquoi enseigner l'informatique ?

Évidence : l'ordinateur est partout !

- Combien d'ordinateurs dans la salle ?
- Combien d'ordinateurs possédez vous ?

Pourquoi enseigner l'informatique ?

Évidence : l'ordinateur est partout !

- Combien d'ordinateurs dans la salle ?
- Combien d'ordinateurs possédez vous ?
- Le mot « assisté par ordinateur » est en voie de disparition

Pourquoi enseigner l'informatique ?

Évidence : l'ordinateur est partout !

- Combien d'ordinateurs dans la salle ?
- Combien d'ordinateurs possédez vous ?
- Le mot « assisté par ordinateur » est en voie de disparition
- Usage constant des ordinateurs, pour le travail comme le reste

Pourquoi enseigner l'informatique ?

Évidence : l'ordinateur est partout !

- Combien d'ordinateurs dans la salle ?
- Combien d'ordinateurs possédez vous ?
- Le mot « assisté par ordinateur » est en voie de disparition
- Usage constant des ordinateurs, pour le travail comme le reste

Évidence : tous les jeunes connaissent déjà l'informatique

Pourquoi enseigner l'informatique ?

Évidence : l'ordinateur est partout !

- Combien d'ordinateurs dans la salle ?
- Combien d'ordinateurs possédez vous ?
- Le mot « assisté par ordinateur » est en voie de disparition
- Usage constant des ordinateurs, pour le travail comme le reste

Évidence : tous les jeunes connaissent déjà l'informatique

Vraiment ?

A. C'est quoi l'informatique

Une petite analogie

- Mr Einstein, vous qui êtes un excellent **physicien**, vous devez savoir changer la roue de ma voiture, non ?

A. C'est quoi l'informatique

Une petite analogie

- Mr Einstein, vous qui êtes un excellent **physicien**, vous devez savoir changer la roue de ma voiture, non ?
- Mr Alonso, vous qui êtes un excellent **conducteur** de F1, vous devez savoir réparer le carburateur de ma voiture, non ?

A. C'est quoi l'informatique

Une petite analogie

- Mr Einstein, vous qui êtes un excellent **physicien**, vous devez savoir changer la roue de ma voiture, non ?
- Mr Alonso, vous qui êtes un excellent **conducteur** de F1, vous devez savoir réparer le carburateur de ma voiture, non ?

Conducteur \neq Garagiste \neq Physicien

A. C'est quoi l'informatique

Une petite analogie

- Mr Einstein, vous qui êtes un excellent **physicien**, vous devez savoir changer la roue de ma voiture, non ?
- Mr Alonso, vous qui êtes un excellent **conducteur** de F1, vous devez savoir réparer le carburateur de ma voiture, non ?

Conducteur \neq Garagiste \neq Physicien

Et pourtant, loin d'être Einstein ou Alonso, ...

- Mr Thiéry, vous qui êtes **professeur en informatique**, vous devez savoir réparer mon W.....s, non ?

C'est quoi l'informatique en fait ?

Suite de la petite analogie ...

C'est quoi l'informatique en fait ?

Suite de la petite analogie ...

L'usage	La technologie	La science
Conduite	Réparation, Conception	Physique

C'est quoi l'informatique en fait ?

Suite de la petite analogie ...

L'usage	La technologie	La science
Conduite	Réparation, Conception	Physique
Consommation	Cuisine	Chimie, Biologie

C'est quoi l'informatique en fait ?

Suite de la petite analogie ...

L'usage	La technologie	La science
Conduite	Réparation, Conception	Physique
Consommation	Cuisine	Chimie, Biologie
Utilisation	Programmation, ...	Informatique

Qu'est-ce qu'on apprend à l'école ?

- Principalement *la science*

Qu'est-ce qu'on apprend à l'école ?

- Principalement *la science*
- Et il y a des raisons profondes pour cela

« Ceux qui sont férus de pratique
sans posséder la science sont comme le pilote
qui s'embarquerait sans timon ni boussole,
et ne saurait jamais avec certitude où il va ».

Léonard de Vinci

Qu'est-ce qu'on apprend à l'école ?

- Principalement *la science*
- Et il y a des raisons profondes pour cela

« Ceux qui sont férus de pratique
sans posséder la science sont comme le pilote
qui s'embarquerait sans timon ni boussole,
et ne saurait jamais avec certitude où il va ».

Léonard de Vinci

- Et il y a des pressions pour que ce ne soit pas le cas ...

Qu'est-ce qu'on apprend à l'école ?

- Principalement *la science*
- Et il y a des raisons profondes pour cela

« Ceux qui sont férus de pratique sans posséder la science sont comme le pilote qui s'embarquerait sans timon ni boussole, et ne saurait jamais avec certitude où il va ».

Léonard de Vinci

- Et il y a des pressions pour que ce ne soit pas le cas ...

Quelle école pour la société de l'information ?

Une conférence de François Élie

À lire ou écouter ... et méditer ...



Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur

Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

La technologie ?

- Qui sait programmer ? Configurer un réseau ?

Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

La technologie ?

- Qui sait programmer ? Configurer un réseau ?

La science ?

Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

La technologie ?

- Qui sait programmer ? Configurer un réseau ?

La science ?

Ma petite expérience

- 6ème :



Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

La technologie ?

- Qui sait programmer ? Configurer un réseau ?

La science ?

Ma petite expérience

– 6ème :



3ème :



Tous les jeunes connaissent déjà l'informatique ?

L'usage ?

- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

La technologie ?

- Qui sait programmer ? Configurer un réseau ?

La science ?

Ma petite expérience

– 6ème :



3ème :



- Fac : apprendre la **science** a chamboulé ma programmation

Tous les jeunes connaissent déjà l'informatique ?

L'usage ?



- Évidence : tous les jeunes savent utiliser un ordinateur
- Vraiment ? ~~les-enfants-ne-savent-pas-se-servir-dun-ordinateur~~

La technologie ?

- Qui sait programmer ? Configurer un réseau ?

La science ?

Ma petite expérience

- 6ème :  3ème : 
- Fac : apprendre la **science** a chamboulé ma programmation
- 2018 : après 30 ans et 300000 lignes de code, j'apprends encore ...

La science informatique ?

- *Science du calcul et de l'information*

La science informatique ?

- *Science du calcul et de l'information*
- Notion fondamentale : **étude des systèmes en évolution**
 - État du système avant
 - Étape de calcul
 - État du système après

La science informatique ?

- *Science du calcul et de l'information*
- Notion fondamentale : **étude des systèmes en évolution**
 - État du système avant
 - Étape de calcul
 - État du système après
- Modèles de calcul

Grands thèmes de l'informatique

Calculabilité : Que peut, ou ne peut pas faire, un ordinateur ?

- Indépendamment du langage
- Indépendamment du matériel
- Miracle : tous les langages sont équivalents !

Grands thèmes de l'informatique

Calculabilité : Que peut, ou ne peut pas faire, un ordinateur ?

- Indépendamment du langage
- Indépendamment du matériel
- Miracle : tous les langages sont équivalents !

Complexité : Combien de ressources pour résoudre un problème ?

- Indépendamment du langage
- Indépendamment du matériel
- Indépendamment de l'algorithme ?

Grands problèmes de l'informatique

Maîtriser les systèmes extrêmement complexes

- Internet avec des milliards d'ordinateurs
- Programmes avec des millions de lignes
- Données occupant des petaoctets (10^{15} octet !)
- Services gérant des millions de clients

Grands problèmes de l'informatique

Maîtriser les systèmes extrêmement complexes

- Internet avec des milliards d'ordinateurs
- Programmes avec des millions de lignes
- Données occupant des petaoctets (10^{15} octet !)
- Services gérant des millions de clients
- Passage à l'échelle

Grands problèmes de l'informatique

Maîtriser les systèmes extrêmement complexes

- Internet avec des milliards d'ordinateurs
- Programmes avec des millions de lignes
- Données occupant des petaoctets (10^{15} octet !)
- Services gérant des millions de clients
- Passage à l'échelle

Abstraction

Exemple : Couches OSI pour les réseaux

Grands problèmes de l'informatique

Maîtriser les systèmes extrêmement complexes

- Internet avec des milliards d'ordinateurs
- Programmes avec des millions de lignes
- Données occupant des petaoctets (10^{15} octet !)
- Services gérant des millions de clients
- Passage à l'échelle

Abstraction

Exemple : Couches OSI pour les réseaux

Difficulté

Apprendre des outils conçus pour les programmes de 100000 lignes en travaillant sur des programmes de 10 lignes ...

Grands thèmes de l'informatique

Concepts des langages de programmation

- Java, C++, Python, Ada, Pascal, Perl, ...
- Un nouveau langage par semaine depuis 50 ans !

Grands thèmes de l'informatique

Concepts des langages de programmation

- Java, C++, Python, Ada, Pascal, Perl, ...
- Un nouveau langage par semaine depuis 50 ans !
- Heureusement les concepts sont presque toujours les mêmes :
 - Programmation impérative
 - Programmation objet
 - Programmation fonctionnelle
 - Programmation logique
 - Algorithmique, Structures de données

Autres grands thèmes de l'informatique

- Architecture des ordinateurs, parallélisme
- Réseaux, transmission de données
- Bases de données
- Langages formels, automates
- Modèles et structures de données
- Sûreté et sécurité du logiciel :
Spécification, Test, Preuve
- Sûreté et sécurité des données :
Codage, cryptographie
- Mathématiques discrètes : graphes, combinatoire, ...

B. À propos de ce cours

Au programme

- **Science** : concepts de la programmation structurée

B. À propos de ce cours

Au programme

- **Science** : concepts de la programmation structurée
- **Technologie** : Programmation C++ (simple) ; Python ?

B. À propos de ce cours

Au programme

- **Science** : concepts de la programmation structurée
- **Technologie** : Programmation C++ (simple) ; Python ?
- **Usage** : Environnement de programmation, GNU/Linux

B. À propos de ce cours

Au programme

- **Science** : concepts de la programmation structurée
- **Technologie** : Programmation C++ (simple) ; Python ?
- **Usage** : Environnement de programmation, GNU/Linux

Ce que l'on va voir

- Les briques de bases, les règles de compositions
- Les constructions usuelles
- Les problèmes déjà résolus, les erreurs les plus courantes

B. À propos de ce cours

Au programme

- **Science** : concepts de la programmation structurée
- **Technologie** : Programmation C++ (simple) ; Python ?
- **Usage** : Environnement de programmation, GNU/Linux

Ce que l'on va voir

- Les briques de bases, les règles de compositions
- Les constructions usuelles
- Les problèmes déjà résolus, les erreurs les plus courantes

Pour quoi faire ?

- Bénéficier de l'expérience de plus de 50 ans de programmation
- Intuition de ce qui est possible ... ou pas
- Intuition de comment résoudre un nouveau problème

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Du TD ? pour quoi faire ???

- Apprendre la science informatique, en utilisant un ordinateur, pour programmer ...

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Du TD ? pour quoi faire ???

- Apprendre la science informatique, en utilisant un ordinateur, pour programmer ...
- Comme apprendre la physique, au volant d'une voiture ...

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Du TD ? pour quoi faire ???

- Apprendre la science informatique, en utilisant un ordinateur, pour programmer ...
- Comme apprendre la physique, au volant d'une voiture ...
- C'est pas facile ...

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Du TD ? pour quoi faire ???

- Apprendre la science informatique, en utilisant un ordinateur, pour programmer ...
- Comme apprendre la physique, au volant d'une voiture ...
- C'est pas facile ...

Une difficulté : la forte hétérogénéité de niveau

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Du TD ? pour quoi faire ???

- Apprendre la science informatique, en utilisant un ordinateur, pour programmer ...
- Comme apprendre la physique, au volant d'une voiture ...
- C'est pas facile ...

Une difficulté : la forte hétérogénéité de niveau

Ce module s'adresse à tous, **débutants** comme **expérimentés**

Organisation du cours

1h30 amphi, 1h45 TD, 2h TP

Du TD ? pour quoi faire ???

- Apprendre la science informatique, en utilisant un ordinateur, pour programmer ...
- Comme apprendre la physique, au volant d'une voiture ...
- C'est pas facile ...

Une difficulté : la forte hétérogénéité de niveau

Ce module s'adresse à tous, **débutants** comme **expérimentés**

Évaluation

- 25% : Partiel (dans l'axe des TD)
- 40% : Examen final (vision d'ensemble)
- 20% : Projet en fin de semestre
- 15% : Exercices en ligne, notes de TP

C. Ordinateurs et traitement automatique des informations

Exemples d'ordinateurs

C. Ordinateurs et traitement automatique des informations

Exemples d'ordinateurs

- Calculatrice (programmable)
- Ordinateur personnel (PC, Mac, ...)
- Station de travail (Sun, DEC, HP, ...)
- Super-ordinateur (Cray, IBM-SP, ...)
- Clusters d'ordinateurs

C. Ordinateurs et traitement automatique des informations

Exemples d'ordinateurs

- Calculatrice (programmable)
- Ordinateur personnel (PC, Mac, ...)
- Station de travail (Sun, DEC, HP, ...)
- Super-ordinateur (Cray, IBM-SP, ...)
- Clusters d'ordinateurs

Mais aussi

- Puce (programme fixe)
- Tablettes
- Téléphones portables, appareils photos, GPS, lecteurs MP3, ...
- Box, routeurs wifi, ...
- Téléviseurs, ...
- Arduino, Raspberry Pi, ...

Caractéristiques principales d'un ordinateur

Caractéristiques principales d'un ordinateur

Absolument stupide

Caractéristiques principales d'un ordinateur

Absolument stupide

- Il obéit strictement aux ordres reçus

Caractéristiques principales d'un ordinateur

Absolument stupide

- Il obéit strictement aux ordres reçus
- Est-ce qu'il fait ce que l'on veut ?

Caractéristiques principales d'un ordinateur

Absolument stupide

- Il obéit strictement aux ordres reçus
- Est-ce qu'il fait ce que l'on veut ?

Très très rapide

Caractéristiques principales d'un ordinateur

Absolument stupide

- Il obéit strictement aux ordres reçus
- Est-ce qu'il fait ce que l'on veut ?

Très très rapide

- 2GHz : 2 milliards d'opérations par seconde

Caractéristiques principales d'un ordinateur

Absolument stupide

- Il obéit strictement aux ordres reçus
- Est-ce qu'il fait ce que l'on veut ?

Très très rapide

- 2GHz : 2 milliards d'opérations par seconde

Très très bonne mémoire

Caractéristiques principales d'un ordinateur

Absolument stupide

- Il obéit strictement aux ordres reçus
- Est-ce qu'il fait ce que l'on veut ?

Très très rapide

- 2GHz : 2 milliards d'opérations par seconde

Très très bonne mémoire

- Bible : Mo (million de caractères)
- Mémoire : Go (milliards de caractères)
- Disque : To (1000 milliards de caractères)
- Data center : Po

À quoi sert un ordinateur ?

Stocker des informations

- Documents, musique, photos, agenda, ...

À quoi sert un ordinateur ?

Stocker des informations

- Documents, musique, photos, agenda, ...

Traiter automatiquement des informations

- *Entrée* : informations venant du clavier, de la souris, de capteurs, de la mémoire, d'autres ordinateurs, ...
- Traitement des informations en exécutant un *programme*
- *Sortie* : information envoyées vers l'écran, la mémoire, d'autres ordinateurs, ...

À quoi sert un ordinateur ?

Stocker des informations

- Documents, musique, photos, agenda, ...

Traiter automatiquement des informations

- *Entrée* : informations venant du clavier, de la souris, de capteurs, de la mémoire, d'autres ordinateurs, ...
- Traitement des informations en exécutant un *programme*
- *Sortie* : information envoyées vers l'écran, la mémoire, d'autres ordinateurs, ...

Définition

Informellement, un *programme* est une séquence d'instructions qui spécifie étape par étape les opérations à effectuer pour obtenir à partir des *entrées* un *résultat* (*la sortie*).

Voir aussi : http://fr.wikipedia.org/wiki/Programme_informatique

Exemples de programmes

Ingrédients

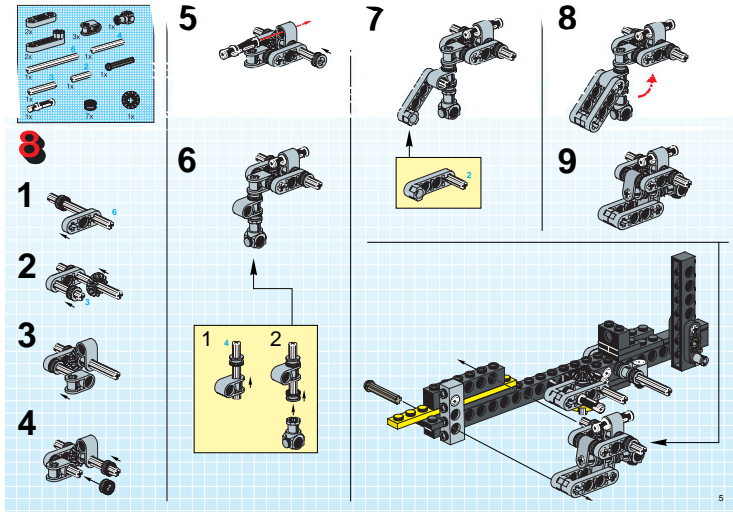
250g de chocolat, 125g de beurre, 6 œufs, 50 g de sucre, café

Étapes

- Faire fondre le chocolat avec 2 cuillères d'eau
- Ajouter le beurre, laisser refroidir puis ajouter les jaunes
- Ajouter le sucre et comme parfum un peu de café
- Battre les blancs jusqu'à former une neige uniforme
- Ajouter au mélange.

Entrée ? Sortie ?

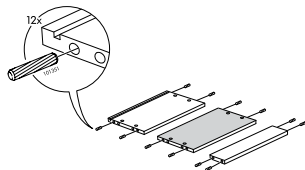
Exemples de programmes



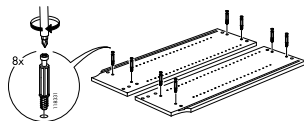
Entrée ? Sortie ?

Exemples de programmes

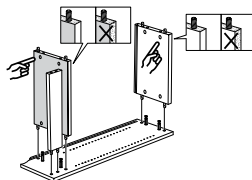
1



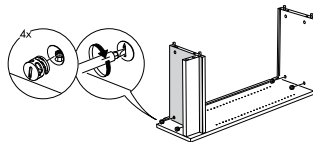
2



3



4



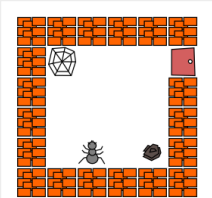
Entrée ? Sortie ?

Exemples de programmes

laby

Démo

Ce labyrinthe sert de démonstration. Le programme initial permet au robot fourmi de sortir de ce labyrinthe. Il va s'amuser un peu avec le caillou et éviter la toile d'araignée. Appuie sur le bouton qui permet de charger le programme dans le robot, et fait défiler le temps avec les flèches.



Language :

Niveau :

Programme :

```
1 #include "robot.h"
2
3 void fourmi()
4 {
5     droite();
6     avance();
7     prend();
8     gauche();
9     avance();
10    pose();
11    droite();
12    avance();
13    gauche();
14    avance();
15    avance();
16    droite();
17    ouvre();
18 }
19
```

Exécuter

Précédent Suivant Rembobiner Lire Avancer

Entrée ? Sortie ?

Un exemple de programme C++

puissance-quatre.cpp

```
#include <iostream>
using namespace std;

int main() {
    int x, xCarre, xPuissanceQuatre;

    cout << "Entrez un entier: ";
    cin >> x;

    xCarre = x * x;
    xPuissanceQuatre = xCarre * xCarre;

    cout << "La puissance quatrième de " << x
         << " est " << xPuissanceQuatre << endl;

    return 0;
}
```

Un exemple de programme C++

puissance-quatre.cpp

```
#include <iostream>
using namespace std;

int main() {
    int x, xCarre, xPuissanceQuatre;

    cout << "Entrez un entier: ";
    cin >> x;

    xCarre = x * x;
    xPuissanceQuatre = xCarre * xCarre;

    cout << "La puissance quatrième de " << x
         << " est " << xPuissanceQuatre << endl;

    return 0;
}
```

Compilation, exécution, ...

Un exemple de programme C++

puissance-quatre.cpp

```
#include <iostream>
using namespace std;

int main() {
    int x, xCarre, xPuissanceQuatre;

    cout << "Entrez un entier: ";
    cin >> x;

    xCarre = x * x;
    xPuissanceQuatre = xCarre * xCarre;

    cout << "La puissance quatrième de " << x
         << " est " << xPuissanceQuatre << endl;

    return 0;
}
```

Compilation, exécution, ... **Un peu lourd ?**

Un exemple de programme C++

puissance-quatre.cpp

```
#include <iostream>
using namespace std;

int main() {
    int x, xCarre, xPuissanceQuatre;

    cout << "Entrez un entier: ";
    cin >> x;

    xCarre = x * x;
    xPuissanceQuatre = xCarre * xCarre;

    cout << "La puissance quatrième de " << x
         << " est " << xPuissanceQuatre << endl;

    return 0;
}
```

Compilation, exécution, ... **Un peu lourd ? Pas de panique !**

Le coeur du programme C++

puissance-quatre-jupyter.cpp

```
// Entrée
int x = 5

// Traitement
int xCarre = x * x;
int xPuissanceQuatre = xCarre * xCarre;

// Sortie
xPuissanceQuatre
```

Le coeur du programme C++

puissance-quatre-jupyter.cpp

```
// Entrée
int x = 5

// Traitement
int xCarre = x * x;
int xPuissanceQuatre = xCarre * xCarre;

// Sortie
xPuissanceQuatre
```

Exécution dans Jupyter+Cling

Jupyter+Cling : une super calculatrice programmable

Jupyter

- Un environnement de calcul interactif multi-langage
- <http://jupyter.org>

Cling

- Un interpréteur C++
- <https://root.cern.ch/cling>

Démo

E. Expressions

Expressions

Combinaison de **valeurs** par des **opérations** donnant une nouvelle **valeur**

Exemples

L'expression $3 * (1 + 3) + (1 + 4) * (2 + 4)$ vaut 42

E. Expressions

Expressions

Combinaison de **valeurs** par des **opérations** donnant une nouvelle **valeur**

Exemples

L'expression $3 * (1 + 3) + (1 + 4) * (2 + 4)$ vaut 42

Opérations sur les entiers

opération	exemple	résultat
opposé	$-(-5)$	5
addition	$17 + 5$	22
soustraction	$17 - 5$	12
multiplication	$17 * 5$	85
division entière	$17 / 5$	3
reste de la division entière	$17 \% 5$	2

Expressions booléennes

Expressions booléennes

Définition (Expression booléenne)

Une expression dont la valeur est **vrai** ou **faux** (type : `bool`)

Expressions booléennes

Définition (Expression booléenne)

Une expression dont la valeur est **vrai** ou **faux** (type : `bool`)

Exemples

`true`

`false`

`x > 3.14`

`2 <= n and n <= 5`

Expressions booléennes

Définition (Expression booléenne)

Une expression dont la valeur est **vrai** ou **faux** (type : **bool**)

Exemples

true

false

$x > 3.14$

$2 \leq n$ **and** $n \leq 5$

Opérations booléennes usuelles

opération	exemple	résultat
comparaison	$3 \leq 5$	true
comparaison	$3 < 5$	true
comparaison	$3 > 5$	false
égalité	$3 == 5$	false
inégalité	$3 != 5$	true
négation	not $3 \leq 5$	false
et	$3 < 5$ and $3 > 5$	false
ou	$3 < 5$ or $3 > 5$	true

Aparté : **syntaxe, sémantique, algorithme**

- **Syntaxe** : comment on l'écrit
- **Sémantique** : ce que cela fait
- **Algorithme** : comment c'est fait

Aparté : **syntaxe**, **sémantique**, **algorithme**

- **Syntaxe** : comment on l'écrit
- **Sémantique** : ce que cela fait
- **Algorithme** : comment c'est fait

Exemple

- Syntaxe : $17 / 5$
- Sémantique : calcule la division entière de 17 par 5
- Algorithme : division euclidienne

F. Variables

Exemple

Calculer l'énergie cinétique $\frac{1}{2}mv^2$ d'un objet de masse 14,5 kg selon qu'il aille à 1, 10, 100, ou 1000 km/h.

Variables

Définition

Une variable est un espace de stockage **nommé** où le programme peut mémoriser une donnée

Le nom de la variable est choisi par le programmeur

Variables

Définition

Une variable est un espace de stockage **nommé** où le programme peut mémoriser une donnée

Le nom de la variable est choisi par le programmeur

- Objectif : stocker des informations durant l'exécution d'un programme
- Analogie : utiliser un récipient pour stocker des ingrédients en cuisine :
 - Verser sucre dans un **saladier**
 - Ajouter la farine dans le **saladier**
 - Laisser reposer ...
 - Verser le contenu du **saladier** dans ...

Variables

Notes

En C++, une variable possède quatre propriétés :

- un *nom* (ou *identificateur*)

Variables

Notes

En C++, une variable possède quatre propriétés :

- un *nom* (ou *identificateur*)
- une *adresse*

Variables

Notes

En C++, une variable possède quatre propriétés :

- un *nom* (ou *identificateur*)
- une *adresse*
- un *type*

Variables

Notes

En C++, une variable possède quatre propriétés :

- un *nom* (ou *identificateur*)
- une *adresse*
- un *type*
- une *valeur*

Variables

Notes

En C++, une variable possède quatre propriétés :

- un *nom* (ou *identificateur*)
- une *adresse*
- un *type*
- une *valeur*

La valeur peut changer en cours d'exécution du programme
(d'où le nom de variable)

Notion de type

Les variables peuvent contenir toutes sortes de données différentes :

- nombres entiers, réels, booléens, ...
- textes
- relevés de notes, images, musiques, ...

Définition (Notion de *type de donnée*)

- Une variable C++ ne peut contenir qu'une seule sorte de données
- On appelle cette sorte le **type** de la variable
- On dit que C++ est un langage typé statiquement

Les types de base

Les différents types de base en C++ sont :

- Les entiers (mots clés `int`, `long int`);

Exemples : 1, 42, -32765

Les types de base

Les différents types de base en C++ sont :

- Les entiers (mots clés `int`, `long int`);

Exemples : 1, 42, -32765

- les réels (mots clés `float` ou `double`);

Exemples : 10.43, 1.0324432e22

Les types de base

Les différents types de base en C++ sont :

- Les entiers (mots clés `int`, `long int`);

Exemples : 1, 42, -32765

- les réels (mots clés `float` ou `double`);

Exemples : 10.43, 1.0324432e22

- les caractères (mot clé `char`);

Exemples : 'a', 'b', ' ', ']'

Les types de base

Les différents types de base en C++ sont :

- Les entiers (mots clés `int`, `long int`);
Exemples : 1, 42, -32765
- les réels (mots clés `float` ou `double`);
Exemples : 10.43, 1.0324432e22
- les caractères (mot clé `char`);
Exemples : 'a', 'b', ' ', ']'
- les chaînes de caractères (mot clé `string`).
Exemples : "bonjour", "Alice aime Bob"

Les types de base

Les différents types de base en C++ sont :

- Les entiers (mots clés `int`, `long int`);
Exemples : 1, 42, -32765
- les réels (mots clés `float` ou `double`);
Exemples : 10.43, 1.0324432e22
- les caractères (mot clé `char`);
Exemples : 'a', 'b', ' ', ']'
- les chaînes de caractères (mot clé `string`).
Exemples : "bonjour", "Alice aime Bob"
- les booléens (mot clé `bool`).
Exemples : `true` (vrai), `false` (faux)

Les types de base

Les différents types de base en C++ sont :

- Les entiers (mots clés `int`, `long int`);
Exemples : 1, 42, -32765
- les réels (mots clés `float` ou `double`);
Exemples : 10.43, 1.0324432e22
- les caractères (mot clé `char`);
Exemples : 'a', 'b', ' ', ']'
- les chaînes de caractères (mot clé `string`).
Exemples : "bonjour", "Alice aime Bob"
- les booléens (mot clé `bool`).
Exemples : `true` (vrai), `false` (faux)

Les entiers, les caractères et les booléens forment les types *ordinaux*

La déclaration des variables

Pour chaque variable, il faut donner au programme son nom et son type.
On dit que l'on *déclare* la variable.

Syntaxe (Déclaration des variables)

```
type nomvariable;  
type nomvariable1, nomvariable2, ...;
```

Exemples

```
int x, y, monEntier;  
float f, g;  
bool b;
```

La déclaration des variables

Pour chaque variable, il faut donner au programme son nom et son type. On dit que l'on *déclare* la variable.

Syntaxe (Déclaration des variables)

```
type nomvariable;  
type nomvariable1, nomvariable2, ...;
```

Exemples

```
int x, y, monEntier;  
float f, g;  
bool b;
```

Note : on ne peut pas redéclarer une variable avec le même nom !

L'affectation

Syntaxe

identificateur = expression;

Exemple

x = 3 + 5;

L'affectation

Syntaxe

identificateur = expression;

Exemple

x = 3 + 5;

Sémantique

- Calcul (ou évaluation) de la valeur de l'expression
- Stockage de cette valeur dans la case mémoire associée à cette variable.
- **La variable et l'expression doivent être de même type !**

Exemples d'affectations

opération	instruction	valeur de la variable après	
affecter la valeur 1 à la variable x	x = 1	x :	1
affecter la valeur 3 à la variable y	y = 3	y :	3

Exemples d'affectations

opération	instruction	valeur de la variable après	
affecter la valeur 1 à la variable x	$x = 1$	x :	1
affecter la valeur 3 à la variable y	$y = 3$	y :	3

Notes

- Affectation $x = y$: copie de la valeur
- L'ancienne valeur de x est perdue !

Exemples d'affectations

opération	instruction	valeur de la variable après	
affecter la valeur 1 à la variable x	$x = 1$	x :	1
affecter la valeur 3 à la variable y	$y = 3$	y :	3

Notes

- Affectation $x = y$: copie de la valeur
- L'ancienne valeur de x est perdue !
- \neq transférer un ingrédient d'un récipient à l'autre

Exemples d'affectations

opération	instruction	valeur de la variable après
affecter la valeur 1 à la variable x	$x = 1$	x : 1
affecter la valeur 3 à la variable y	$y = 3$	y : 3

Notes

- Affectation $x = y$: copie de la valeur
- L'ancienne valeur de x est perdue !
- \neq transférer un ingrédient d'un récipient à l'autre

opération	instruction	valeur de la variable après
affecter la valeur $x + 1$ à la variable x	$x = x + 1$	x : 2
affecter la valeur $y + x$ à la variable y	$y = y + x$	y : 5

Affectation et égalité : deux concepts différents

Exemple

Exécution répétée de $x = x + 1$

Affectation et égalité : deux concepts différents

Exemple

Exécution répétée de $x = x + 1$

L'affectation $x = 5$

Une instruction modifiant l'état de la mémoire.

Le test d'égalité $x == 5$

Une expression qui a une valeur booléenne (vrai ou faux) :

« Est-ce que x est égal à 5 ? »

Autrement dit : est-ce que la valeur contenue dans la variable x est 5 ?

Fonctions

Retour sur

Exemple

Calculer l'énergie cinétique $\frac{1}{2}mv^2$ d'un objet de masse 14,5 kg selon qu'il aille à 1, 10, 100, ou 1000 km/h.

Fonctions

Retour sur

Exemple

Calculer l'énergie cinétique $\frac{1}{2}mv^2$ d'un objet de masse 14,5 kg selon qu'il aille à 1, 10, 100, ou 1000 km/h.

Comment éviter de retaper chaque fois la formule ?

Exemple sur Jupyter

Fonctions

Définition informelle

Une fonction est un petit programme :

- Entrées
- Traitement
- Sortie

Fonctions

Définition informelle

Une fonction est un petit programme :

- Entrées
- Traitement
- Sortie

Exemple

```
float energie_cinetique(float m, float v) {  
    return 0.5 * m * v * v;  
}
```

- Entrées : la masse et la vitesse (des nombres réels)
- Sortie : l'énergie cinétique (un nombre réel)
- Traitement : $0.5 * m * v * v$

Résumé

- À propos d'Info 111
 - Qu'est-ce que l'informatique (**Usage, Technologie, Science !**)

Résumé

- À propos d'Info 111
 - Qu'est-ce que l'informatique (**Usage, Technologie, Science !**)
 - Objectifs du cours

Résumé

- À propos d'Info 111
 - Qu'est-ce que l'informatique (**Usage, Technologie, Science !**)
 - Objectifs du cours
 - Un aperçu des premiers éléments de programmation :
 - Programmes
 - Expressions
 - Types
 - Variables (**Affectation \neq Égalité !**)
 - Fonctions
- On reviendra dessus !

Résumé

- À propos d'Info 111
 - Qu'est-ce que l'informatique (**Usage, Technologie, Science !**)
 - Objectifs du cours
- Un aperçu des premiers éléments de programmation :
 - Programmes
 - Expressions
 - Types
 - Variables (**Affectation \neq Égalité !**)
 - Fonctions

On reviendra dessus !
- Environnement Jupyter+Cling