Introduction à la programmation

Patrick Fuchs (Pierre Poulain)

DUBii (01/2019→02/2019)

université PARIS DIDEROT PARIS 7


Programmer ?

```
allAtom=Rigidbody(atomicName)
sys.stderr.write("Load atomic file %s with %d atoms \n" %(atomicName, allAtom.Size()))

#extract all 'atoms' objects
atomList=[]
for i in xrange(allAtom.Size()):
        atom = allAtom.CopyAtom(i)
        # look for residue or base type conversion
        resName = atom.GetResidType()
        if resName in resConv.keys():
                atom.SetResidType( resConv[resName] )
        # look for atom type conversion
        atomTag = atom.GetResidType() + '-' + atom.GetType()
        if atomTag in atomConv.keys():
                atomName = atomConv[atomTag].split('-')[1]
                atom.SetType( atomName )
        atomList.append(atom)

#count residues
residueTagList=[]
coarseResList=[]
for atom in atomList:
        resName = atom.GetResidType()
        # create a unique identifier for every residue
        # resTag is for instance "LEU-296-A"
        resTag = resName + '-'+ str(atom.GetResidId()) + '-' + atom.GetChainId()
        if resTag not in residueTagList:
                if resBeadAtomModel.has_key(resName):
                        residueTagList.append(resTag)
                        # add a pattern residue to the list of coarse residues for the protein
                        # beware of the hugly list copy: use copy.deepcopy() !
                        coarseResList.append(copy.deepcopy(resBeadAtomModel[resName]))
                else:
                        sys.stderr.write("WARNING: residue %s is unknown the residues <-> beads <-> atom
                        sys.stderr.write("       : residue %s will not be reduced into coarse grain\n" %
sys.stderr.write("Number of residues: %i\n" %(len(residueTagList)))
```

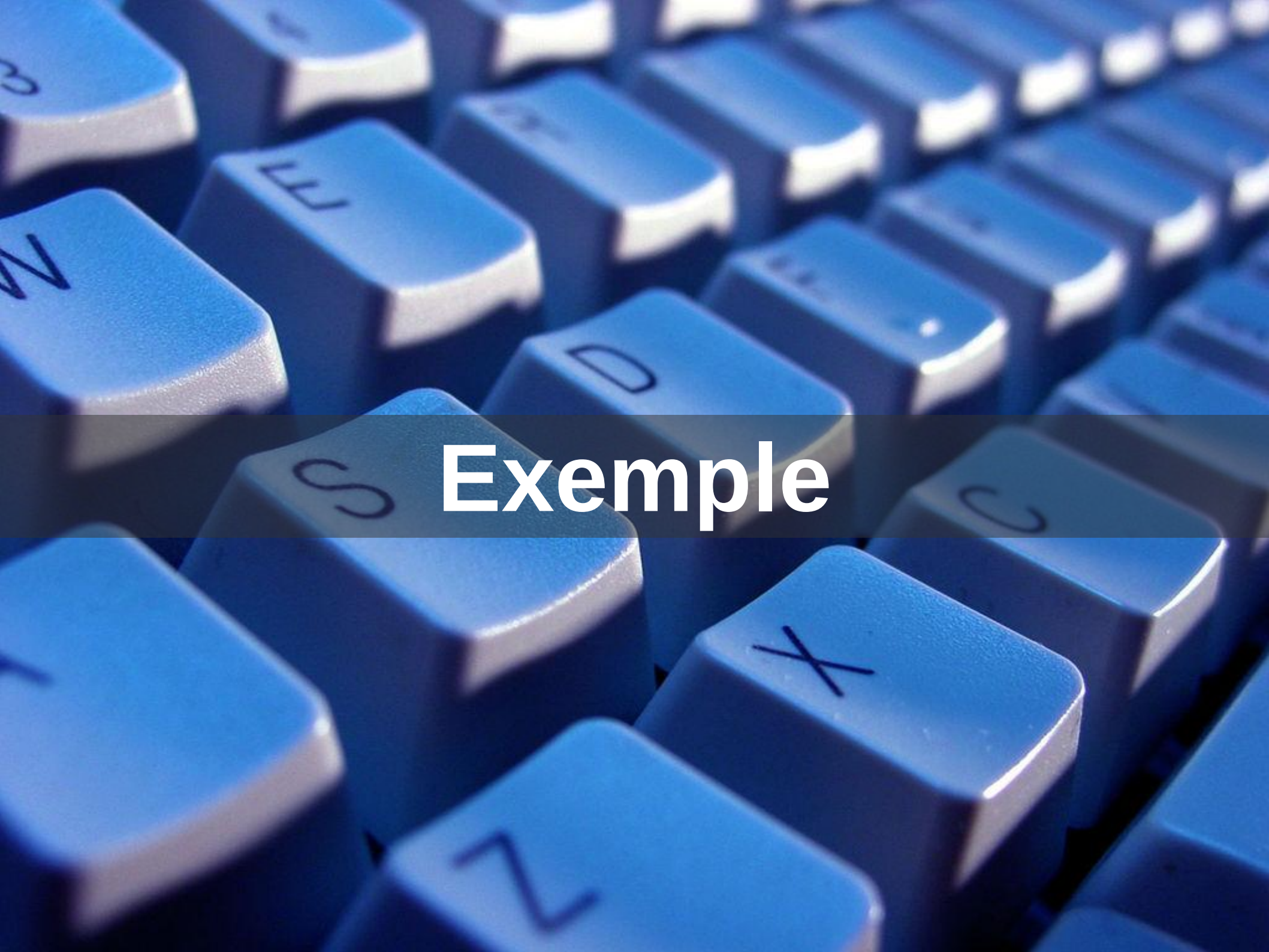
Donner des ordres


Pourquoi ?


Stocker


Trier


Répéter


Exemple

Combien y a-t-il d'alanines dans cette protéine ?

GWGAWILAGAGA

Combien y a-t-il d'alanines dans cette protéine ?

GWGAWILAGAGA

1 2 3 4

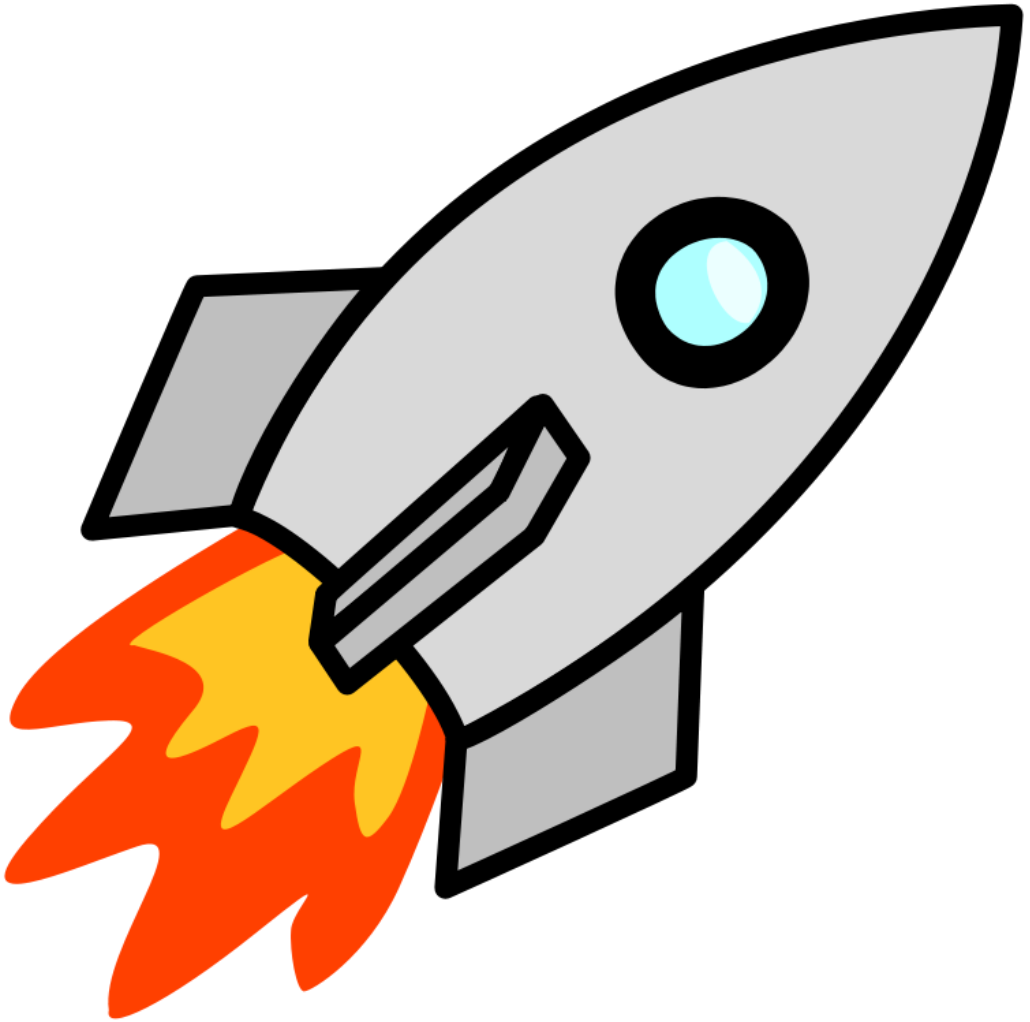
Et dans celle-ci ?

MRARPRPRPLWATVLALGALAGVGVGGPNICTTRGVSSCQQCLAVSPMCAWCSDEALPLG
SPRCDLKENLLKDNCAPESIEFPVSEARVLEDRPLSDKGSGDSSQVTQVSPQRIALRLRP
DDSKNFSIQVRQVEDYPVDIYYLMDLSYSMKDDLWSIQNLGTKLATQMRKLTSNLRIGFG
AFVDKPVSPYMYISPPEALENPCYDMKTTCLPMFGYKHVLTLTDQVTRFNEEVKKQSVSR
NRDAPEGGFDAIMQATVCDEKIGWRNDASHLLVFTTDAKTHIALDGRLAGIVQPNDGQCH
VGSDNHYSASTTMDYPSLGLMTEKLSQKNINLIFAVTENVVNLYQNYSELIPGTTVGVLS
MDSSNVLQLIVDAYGKIRSKVELEVRDLPEELSLSFNATCLNNEVIPGLKSCMGLKIGDT
VSFSIEAKVRGCPQEKEKSFTIKPVGFKDSLIVQVTFDCDCACQAQAEPNSHRCNNGNGT
FECGVCRCGPGWLGSQCECSEEDYRPSQQDECSPREGQPVCSQRGECLCGQCVCHSSDFG
KITGKYCECDDFSCVRYKGEMCSGHGQCSCGDCLCDSDWTGYYCNCTTRTDTCMSSNGLL
CSGRGKCECGSCVCIQPGSYGDTCEKCPTCPDACTFKKECVECKKFDRGALHDENTCNRY
CRDEIESVKELKDTGKDAVNCTYKNEDDCVVRFQYYEDSSGKSILYVVEEPECPKGPDIL
VVLLSVMGAILLIGLAALLIWKLLITIHDRKEFAKFEEERARAKWDTANNPLYKEATSTF
TNITYRGT

Glycoprotéine plaquettaire humaine β3


Un ordinateur

très rapide



beaucoup de mémoire

mais bête

décomposer

un problème complexe


en éléments simples


Algorithme

Notion d'algorithme

Wikipedia: Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème.

Problème:

J'ai 20 € dans ma poche, je veux m'acheter Linux mag, et ensuite j'aimerais aller au ciné, sinon je mangerai une glace.

Algorithme:

```
prix_place_de_ciné <- 11
prix_linux_mag <- 4.9
budget <- 20
# j'achete Linux mag
budget <- budget - prix_linux_mag
# je vais au ciné
si prix_place_de_ciné < budget:
   je vais au ciné
sinon:
   j'achète une glace avec le reste du budget
```

Algorithme humain

GWGAWILAGAGA

1 2 3 4

Pour chaque acide aminé de la séquence,

si l'acide aminé est A

alors on compte une alanine de plus.

Algorithme humain

```
GWGAWILAGAGA
   1    2 3 4
```

Pour chaque acide aminé de la séquence,

si l'acide aminé est A

alors on compte une alanine de plus.

Algorithme Python

```
sequence = "GWGAWILAGAGA"
nombre_ala = 0

for acide_amine in sequence:
    if acide_amine == "A":
        nombre_ala = nombre_ala + 1

print(nombre_ala)
```


Autres langages ?

Java

```
package comptagealanines;
public class Main {
  public static void main(String[] args) {

    String sequence = "GWGAWILAGAGA";
    int nombre_ala = 0 ;

    for (int i = 0 ; i < sequence.length(); i++ ){
      if (sequence.charAt( i ) == 'A'){
        nombre_ala = nombre_ala + 1;
      }
    }

    System.out.println(nombre_ala);
  }
}
```

Perl

```
my @sequence = split('','GWGAWILAGAGA');
my $nombre_ala = 0 ;

foreach my $acide_amine (@sequence)
{
  if ($acide_amine eq 'A')
  {
    $nombre_ala = $nombre_ala + 1;
  }
}

print "$nombre_ala\n";
```

Javascript

```
<script>
var sequence = "GWGAWILAGAGA";
var nombre_ala = 0;

for(i=0; i<sequence.length; i++) {
      if(sequence[i].toUpperCase() == "A") {
            nombre_ala = nombre_ala + 1;
      }
}

window.alert("Nombre de A : " + nombre_ala);
</script>
```

PHP

```php
<?php
	$sequence = "GWGAWILAGAGA";
	$nombre_ala = 0;

	for($i=0; $i<strlen($sequence); $i++) {
		if(strtoupper($sequence[$i]) == "A") {
			$nombre_ala = $nombre_ala + 1;
		}
	}
	echo "Nombre de A : ".$nombre_ala;
?>
```

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            string sequence = "GWGAWILAGAGA";
            int nombre_ala = 0;
            for (int i = 0; i < sequence.Length; i++)
            {
                if (sequence[i].ToString().ToUpper() == "A")
                {
                    nombre_ala = nombre_ala + 1;
                }
            }
            Console.WriteLine("Nombre de A : " + nombre_ala.ToString());
        }
    }
}
```


Points communs

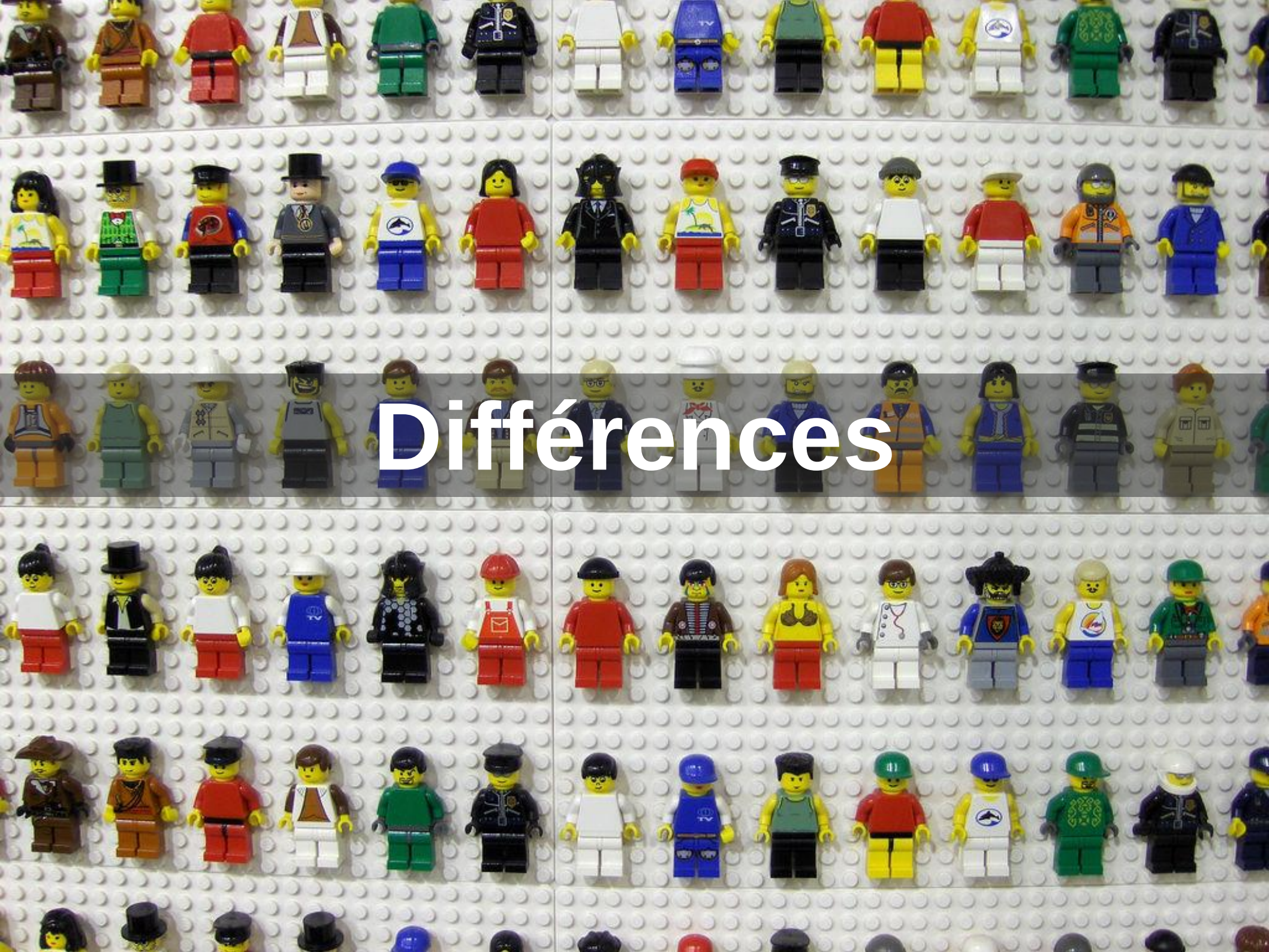
```
sequence = "GWGAWILAGAGA"
nombre_ala = 0

for acide_amine in sequence:
        if acide_amine == "A":
                nombre_ala = nombre_ala + 1

print nombre_ala
```

Variables

Données

Tests / Boucles

Opérations


Différences

; { } @ $

using public class
static void

Compilation

C, C++, C#, Fortran

code source → langage machine → programme exécutable

Interprétation (*bytecode*)

Java, Python, Perl

code source → *bytecode* (caché)

Interprétation

Bash, Javascript, PHP

code source
(ligne par ligne)

Apprentissage de la programmation

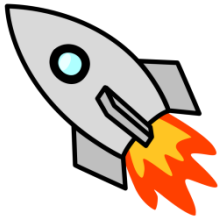
Crédits graphiques

PPDIGITAL (Flickr)

Katherine Donaldson (Flickr)

Ralphbijker (Flickr)

Olivcris (Flickr)



Nicobunu (Openclipart.org)

VisualPharm (Findicons)

TurboMilk (Findicons)

Wilsoninc (Findicons)

fe2cruz (Flickr)

713 Avenue (Flickr)

K Lee (Wikipedia)

selva (Flickr)