

Introduction à la programmation

Patrick Fuchs (Pierre Poulain)

DUBii (01/2019→02/2019)



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 <-> atoms\n" % resName)
            sys.stderr.write("      : residue %s will not be reduced into coarse grain\n" % resName)
sys.stderr.write("Number of residues: %i\n" %(len(residueTagList)))
```

Donner des ordres

A close-up photograph of a computer keyboard with a strong blue color cast. The keys are arranged in a standard QWERTY layout. Visible characters include 'W', 'E', 'D', 'S', 'C', 'X', and 'Z'. The lighting creates highlights on the edges of the keys, giving them a three-dimensional appearance.

Pourquoi ?

A photograph of a filing cabinet filled with numerous wooden index cards. The cards are organized into rows and columns, with many featuring colorful labels (red, green, blue, yellow) and some with handwritten text. A semi-transparent dark band runs horizontally across the center of the image, containing the word "Stocker" in a large, white, sans-serif font. The background shows the wooden structure of the filing cabinet and the dense arrangement of the cards.

Stocker

A close-up photograph of a child's hand, wearing a blue and purple long-sleeved shirt, sorting small, colorful sticks into a dark grey muffin tin. The tin has several compartments, each containing a white paper liner. The sticks are in various colors: green, yellow, orange, red, dark red, black, and pink. The child's hand is currently placing a light blue stick into one of the compartments. The background is a light-colored, textured surface, possibly a carpet.

Trier



Répéter

A close-up photograph of a computer keyboard with a strong blue color cast. The keys are arranged in a standard QWERTY layout. Visible characters include 'W', 'E', 'D', 'S', 'X', and 'Z'. The lighting creates highlights on the edges of the keys, giving them a three-dimensional appearance.

Exemple

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

GWGAWILAGAGA

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

GWGAWIILAGAGA

1

2

3

4

Et dans celle-ci ?

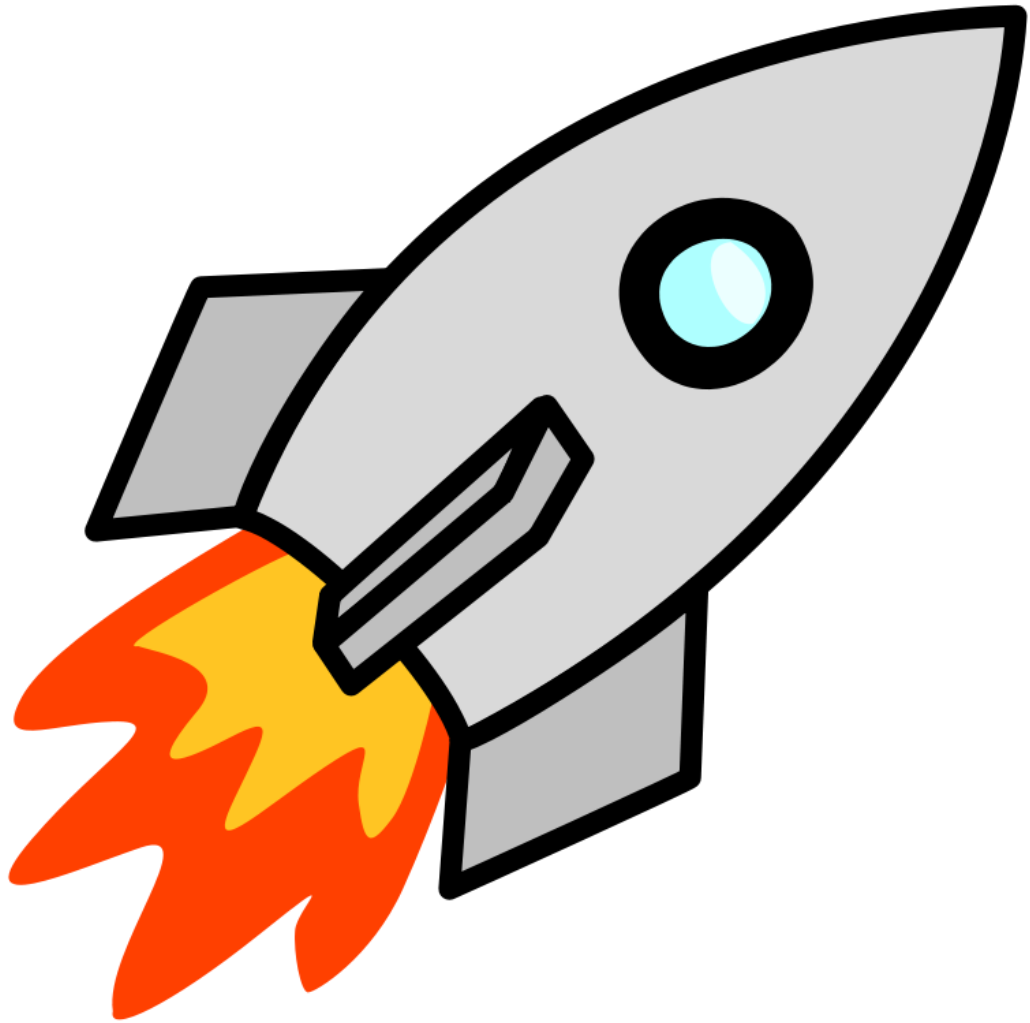
MRARPRPRPLWATVLAALGALAGVGVGGPNICTTRGVSSCQQCLAVSPMCAWCSDEALPLG
SPRCDLKENLLKDNCAPESEIEFPVSEARVLEDRLSDKSGDSSQVTQVSPQRIALRLRP
DDSKNFSIQVRQVEDYPVDIYYLMDLSYSMKDDLWSIQNLGTKLATQMRKLTSNLRIGFG
AFVDKPVSPYMYISPPEALENPCYDMKTTCLPMFGYKHVLTLLTDQVTRFNEEVKKQSVSR
NRDAPEGGFDAIMQATVCDEKIGWRNDASHLLVFTTDAKTHIALDGRLAGIVQPNDGQCH
VGSDNHYSASTTMDYPSLGLMTEKLSQKNINLIFAVTENVVNLYQNYSELIPGTTVGVL
MDSSNVLQLIVDAYGKIRSKVELEVRDLPEELSLSFNATCLNNEVIPGLKSCMGLKIGDT
VSFSIEAKVRGCPQEKEKSFTIKPVGFKDSLIVQVTFDCDCACQAQAEPNSHRCNNGNGT
FECGVCRCGPGWLGSQCECSEEDYRPSQQDECSREGQPVCSQRGECLCGQCVCHSSDFG
KITGKYCECDDFSCVRYKGEMCSGHGQCSCGDCLCDSDWTGYCNCCTTRTDTCMSSNGLL
CSGRGKCECGSCVCIQPGSYGDTCEKCPTCPDACTFKKECVECKKFDRGALHDENTCNRY
CRDEIESVKELKDTGKDAVNCTYKNEDDCVVRFQYYEDSSGKSILYVVEEPEC PKGPDIL
VLLSVMGAILLIGLAALLIWKLLITIHDRKEFAKFEEERARAKWDTANNPLYKEATSTF
TNITYRGT

Glycoprotéine plaquettaire humaine $\beta 3$

A close-up photograph of a computer keyboard, heavily tinted with a blue color. The keys are arranged in a standard QWERTY layout. Visible characters include 'W', 'E', 'D', 'C', 'X', and 'Z'. The lighting creates highlights on the edges of the keys, giving them a three-dimensional appearance. A semi-transparent dark blue horizontal band is positioned across the middle of the image, serving as a background for the text.

Un ordinateur

très rapide





**beaucoup
de mémoire**

mais bête



décomposer



un problème complexe



en éléments simples

A close-up photograph of a computer keyboard with a strong blue color cast. The keys are slightly raised and have various characters printed on them, including 'W', 'E', 'D', 'S', 'C', 'X', and 'Z'. A semi-transparent dark blue horizontal band is overlaid across the middle of the image, containing the word 'Algorithme' in white text.

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
```

```
    $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;
```

```
?>
```



```
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

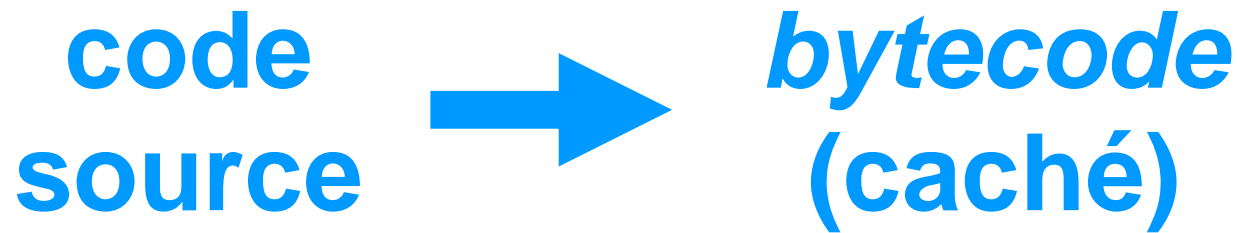
Compilation

C, C++, C#, Fortran



Interprétation (*bytecode*)

Java, Python, Perl



Interprétation

Bash, Javascript, PHP

**code source
(ligne par ligne)**

A close-up photograph of a computer keyboard with a strong blue color cast. The keys are slightly out of focus, with some characters like 'V', 'F', 'D', 'S', 'C', 'X', and 'Z' visible. A semi-transparent dark blue horizontal band is overlaid across the middle of the image, containing the title text in white.

Apprentissage de la programmation

Crédits graphiques



PPDIGITAL (Flickr)



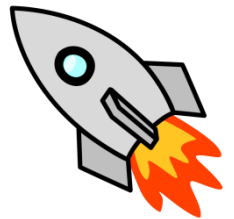
Katherine Donaldson (Flickr)



Ralphbijker (Flickr)



Olivcris (Flickr)



Nicobunu (Openclipart.org)



VisualPharm (Findicons)



TurboMilk (Findicons)



Wilsoninc (Findicons)



fe2cruz (Flickr)



713 Avenue (Flickr)



K Lee (Wikipedia)



selva (Flickr)