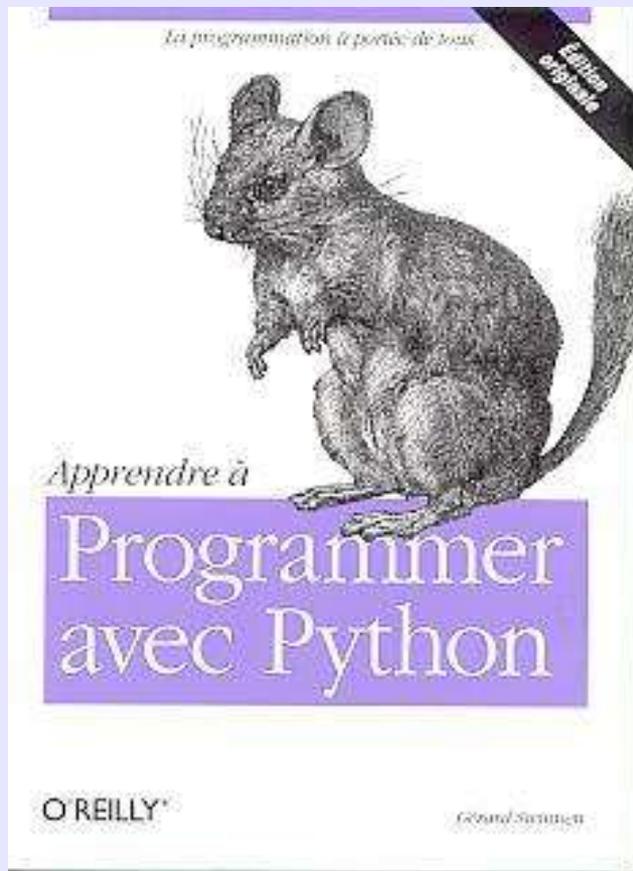


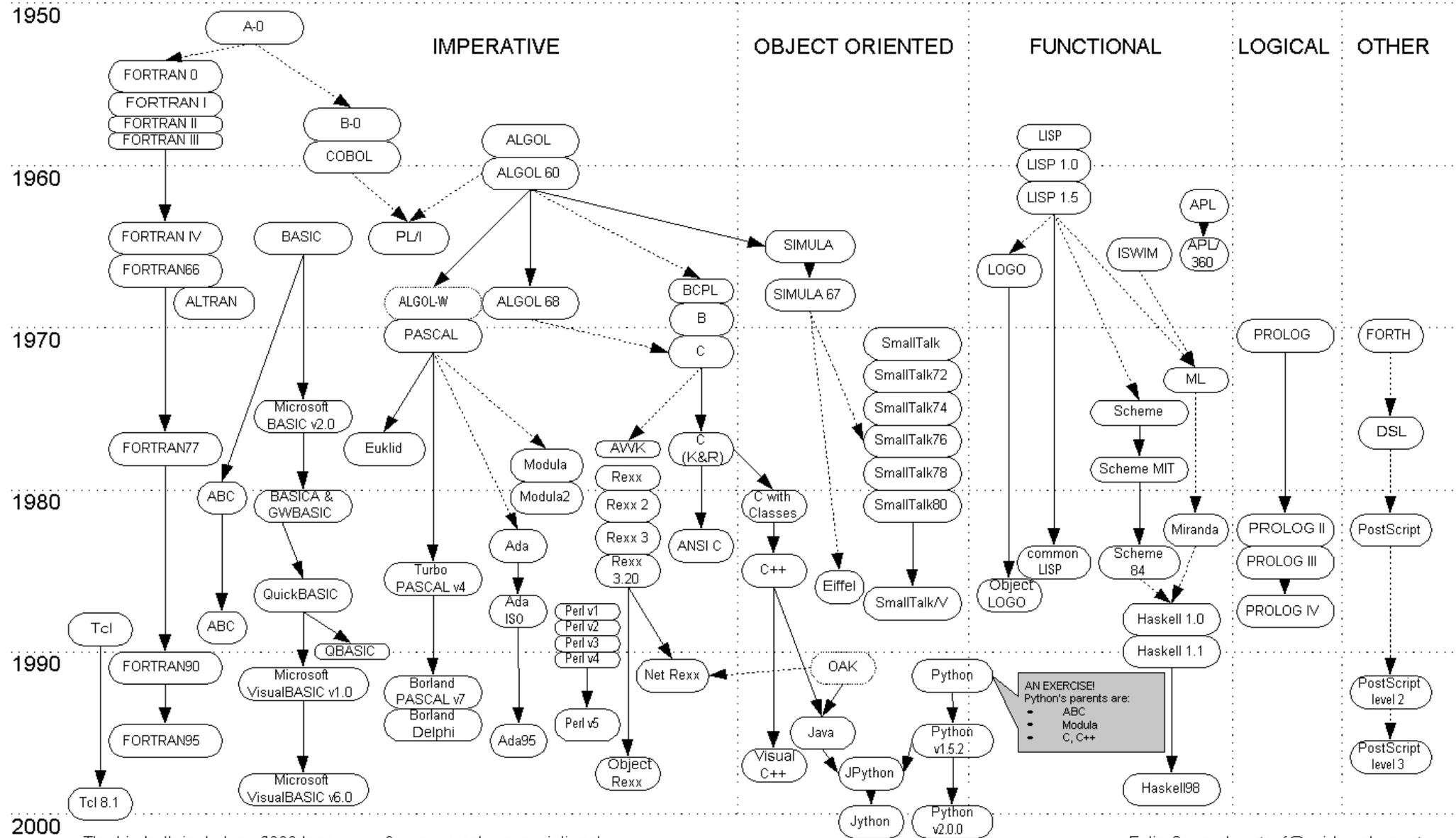
Apprendre à programmer avec Python



Gérard Swinnen
Professeur & Conseiller pédagogique
Institut St. Jean-Berchmans, Liège
Libr' East, Avril 2004

COMPUTER LANGUAGES EVOLUTION

GENEALOGY TREE BETWEEN 1950-2000



The big truth includes ~2000 languages & very complex associations!
 Read for more: <http://perso.wanadoo.fr/levenez/lang/history.html>

<http://amstel.science.uva.nl/~fotisg/python>

Fotis Georgatos <gef@ceid.upatras.gr>
 Amstel Institute, Amsterdam, June 2002

Pourquoi Python ?

- Niveau d'abstraction du langage
- Liberté d'utilisation et de distribution (Licence LGPL)
- Indépendance vis-à-vis de l'O.S. (Portabilité)
- Excellente lisibilité (Syntaxe, structuration)
- Types de données intéressantes (Listes, dictionnaires)
- Extensibilité, Adaptabilité (Modules)
- Variété des approches (Progr. procédurale, objets)

Niveau d'abstraction élevé

- Faut-il, dans l'enseignement secondaire, former déjà des programmeurs professionnels, ou bien préparer les citoyens de demain à vivre dans un univers technologique qui contient de jour en jour davantage d'éléments programmables ?
- La programmation est une tâche ardue. Faut-il la rendre plus difficile encore, en y ajoutant des obstacles artificiels ?
- Faut-il pour autant que l'enseignement utilise d'autres outils que ceux des professionnels ?

Liberté d'utilisation

- *Lesser GNU Public License :*

Les outils Python peuvent être utilisés sans restriction pour produire des programmes de tous types, même si ceux-ci sont distribués ensuite avec une licence plus restrictive que la GPL.

- Le code source de Python lui-même (interpréteur, modules, etc.) est disponible et modifiable.

Portabilité

- Faire travailler effectivement les élèves sur différents systèmes d'exploitation, mais avec des outils similaires → meilleure appropriation des invariants de l'informatique.
- Faire percevoir l'importance des normes et des standards

Lisibilité : le problème

```
extern int
    errno
    ;char
        grrr
            r,
                ;main(
                    int argc
                        char *argv[];{int
                            j,cc[4];printf("
                                choo choo\n"
                                    ) ;
                                        | cc[
                                            !
                                                j ]
                                                    & P(j
                                                        )>2 ?
                                                            j
                                                                :
                                                                    i ){*
                                                                        argv[i++
                                                                            +!-i]
                                                                                };
                                                                                    for
                                                                                        (i=
                                                                                            0;;
                                                                                                i++
                                                                                                    );
                                                                                                        _exit(argv[argc-
                                                                                                            2
                                                                                                                / cc[1*argc]|-1<<4
                                                                                                                    ]
                                                                                                                        ) ;printf("%d",P(""));}}
                                                                                                        P
                                                                                                            (
                                                                                                                a
                                                                                                                    )
                                                                                                                        char a
                                                                                                                            ; {
                                                                                                                                a
                                                                                                                                    ;
                                                                                                                                        while(
                                                                                                                                            a
                                                                                                                                                >
                                                                                                                                                    "
                                                                                                                                                        B
                                                                                                                                                            "
                                                                                                                    /*
                                                                                                                        -
                                                                                                                            by E
                                                                                                                                ricM
                                                                                                                                    arsh
                                                                                                                                        all-
                                                                                                                                            */);
                                                                                                                        }
```

Programme C écrit par Eric Marshall en 1986 :
ce programme affiche "choo choo".

Lisibilité

Exemple de code Python :

```
# Elimination de doublons dans une liste :  
  
lst1 = [9, 12, 40, 5, 12, 3, 27, 5, 9, 3, 8, 22, 40, 3, 6, 25]  
lst2 = []  
  
for elem in lst1:  
    if elem not in lst2:  
        lst2.append(elem)  
  
lst2.sort()  
print lst2
```

Syntaxe simple et intuitive : avantages

- Apprentissage aisé et rapide
(permettant le développement précoce de projets)
- Maintenance du code facilitée
(ainsi que les corrections du professeur !)
- Moins d'erreurs de syntaxe
- Faible fréquence de consultation de la doc.

Syntaxe simple et intuitive

Exercice : afficher tous les nombres entiers de 0 à 9999

Programme Java :

```
public class ConsoleTest {  
    public static void main(string[] args) {  
        for (int i = 0; i < 10000; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

Programme Python :

```
for x in range(10000):  
    print x
```

Niveau d'abstraction des structures de données

Les listes :

```
lst1 = []
```

```
lst2 = [3, 5, 7, 11, 17]
```

```
lst3 = ["Olga", "Marcel", "Zoé"]
```

```
lst4 = [1492, 3.1416, "Sophie"]
```

```
lst5 = [["p", "q", "y"], [5, 37, 19, 23]]
```

Niveau d'abstraction des structures de données

Les dictionnaires :

```
dct1 = {}
```

```
dct2 = {"a": 3, "x": 5, "z": 47}
```

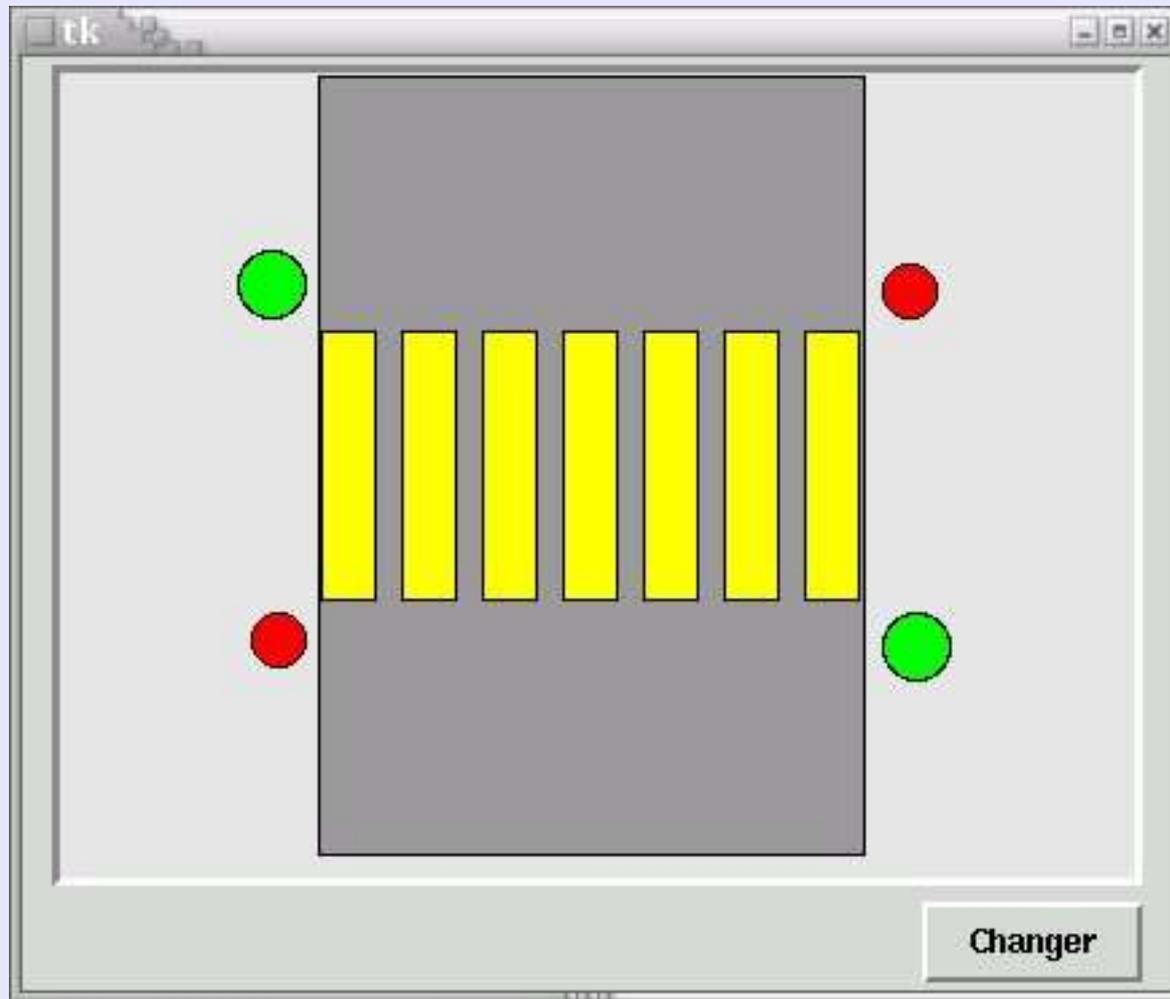
```
dct3 = {5: "rouge", 8: "bleu", 3: "vert"}
```

```
dct4 = {6: ["p", "q"], 4: [5, 37, 19, 23]}
```

```
dct5 = {[63, 17]: "pin", [42, 89]: "chêne"}
```

Exemple de développement

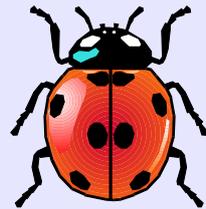
Cahier des charges :



Méthodologie

- Application graphique → objectif très concret, attrayant
- Utilisation de bibliothèques d'objets existants → familiarité avec les concepts de méthode et d'attribut
- Lorsqu'un problème devient complexe, le fragmenter en sous-problèmes plus simples
- Utiliser les structures de données de niveau élevé
- Aborder la programmation de classes dès que possible

Ressources didactiques G. Swinnen :



<http://www.ulg.ac.be/cifen/inforef/swi>