

La transition Scratch/Python

Lionel Vaux Auclair

Groupe *Enseignement de l'informatique*, IREM d'Aix-Marseille
& I2M, université d'Aix-Marseille

5e journée académique sur l'enseignement de l'informatique
7 avril 2021

Au collège

Le programme de cycle 4

[https://eduscol.education.fr/90/j-enseigne-au-cycle-4:](https://eduscol.education.fr/90/j-enseigne-au-cycle-4)

- ▶ en maths (pp. 136), c'est succinct
- ▶ en techno (pp. 122/124) c'est ambitieux (et « boucle conditionnelle », c'est ambigu)

Au collège

Le programme de cycle 4

<https://eduscol.education.fr/90/j-enseigne-au-cycle-4>:

- ▶ en maths (pp. 136), c'est succinct
- ▶ en techno (pp. 122/124) c'est ambitieux (et « boucle conditionnelle », c'est ambigu)

Les attendus en math:

- ▶ c'est trois fois la même chose pour algo/prog, sauf le niveau d'attendus (1 à 3), résumé dans les repères;
- ▶ au niveau 3, il faut savoir utiliser des boucles, accumuler des résultats dans des variables, utiliser les événements, les blocs personnalisés, ...

Au lycée: voie générale

Dans le cours de math de 2nde:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138131

Au lycée: voie générale

Dans le cours de math de 2nde:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138131
- ▶ un peu d'algo partout
- ▶ gros morceau: la notion de fonction

Au lycée: voie générale

Dans le cours de math de 2nde:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138131
- ▶ un peu d'algo partout
- ▶ gros morceau: la notion de fonction

En SNT:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138143

Au lycée: voie générale

Dans le cours de math de 2nde:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138131
- ▶ un peu d'algo partout
- ▶ gros morceau: la notion de fonction

En SNT:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138143
- ▶ *notions transversales de programmation* \simeq cycle 4 + fonctions
- ▶ référence explicite au cours de math

Au lycée: voie générale

Dans le cours de math de 2nde:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138131
- ▶ un peu d'algo partout
- ▶ gros morceau: la notion de fonction

En SNT:

- ▶ https://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=138143
- ▶ *notions transversales de programmation* \simeq cycle 4 + fonctions
- ▶ référence explicite au cours de math

En première:

- ▶ en math: les listes;
- ▶ en NSI: tout.

Au lycée: voie professionnelle

<https://eduscol.education.fr/94/j-enseigne-au-lycee-professionnel>

CAP:

- ▶ attendus très modestes;
- ▶ on reste sur Scratch.

Au lycée: voie professionnelle

<https://eduscol.education.fr/94/j-enseigne-au-lycee-professionnel>

CAP:

- ▶ attendus très modestes;
- ▶ on reste sur Scratch.

Bac pro (nouvelle terminale à la rentrée 2021)

- ▶ algorithmique et programmation sont travaillés dans l'étude des différents domaines du programme (pas de cours spécifique);
- ▶ on programme en Python (même si la notion de « types de variable » n'a pas de sens en Python);
- ▶ on introduit la notion de *fonction*.

Similitudes

Scratch et Python sont tous les deux des langages:

- ▶ séquentiels;
- ▶ impératifs;
- ▶ interprétés.

Dictionnaire bilingue: démarrage

quand  est cliqué

début du programme
(F5 dans IDLE, la flèche verte dans Pyscripter, l'appel à Python depuis la ligne de commande, ...)

Dictionnaire bilingue: les variables



`x = valeur`



`x += valeur`

ou bien : `x = x+valeur`

Dictionnaire bilingue: entrées/sorties

dire

```
print("bonjour")
```

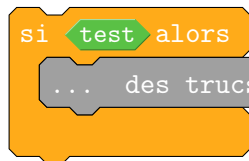
demander et attendre

mettre à

```
nom=input("Quel est votre nom ? ")
```

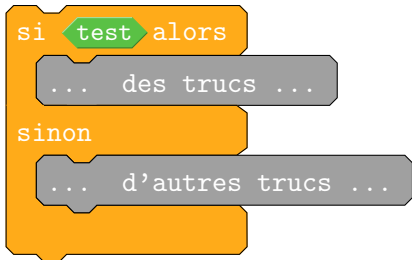
helloworld

Dictionnaire bilingue: les conditionnelles



```
if test:  
    # ... des trucs ...
```

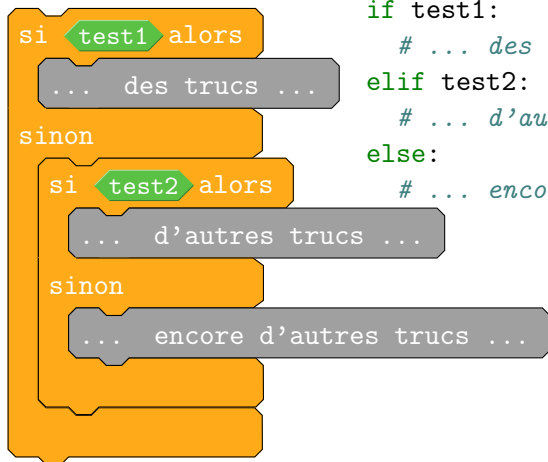

Dictionnaire bilingue: les conditionnelles



```
if test:  
    # ... des trucs ...  
else:  
    # ... d'autres trucs ...
```

isocele

Dictionnaire bilingue: les conditionnelles



```
if test1:  
    # ... des trucs ...  
elif test2:  
    # ... d'autres trucs ...  
else:  
    # ... encore d'autres trucs ...
```

isocele (mieux)

Similitudes

Scratch et Python sont tous les deux des langages:

- ▶ séquentiels;
- ▶ impératifs;
- ▶ interprétés.

En plus:

- ▶ la séquentialité = le passage à la ligne suivante;

Similitudes

Scratch et Python sont tous les deux des langages:

- ▶ séquentiels;
- ▶ impératifs;
- ▶ interprétés.

En plus:

- ▶ la séquentialité = le passage à la ligne suivante;
- ▶ l'indentation en Python colle bien avec la présentation des sous-blocs en Scratch;

Mais...

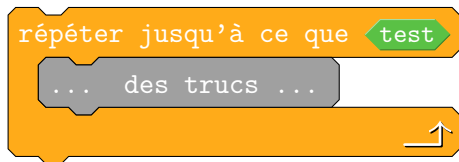
en Python

les valeurs ont un type unique
(int, float, str, ...)

en Scratch

en Scratch les conversions sont
implicites

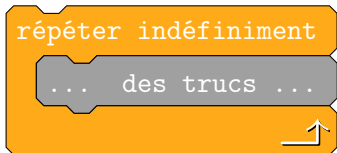
Dictionnaire bilingue: les boucles



```
while not test:  
    # ... des trucs ...
```

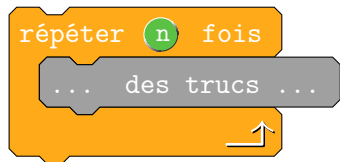
La condition est inversée!

Dictionnaire bilingue: les boucles



```
while True:  
    # ... des trucs ...
```

Dictionnaire bilingue: les boucles



```
for i in range(n):  
    # ... des trucs ...
```

Dictionnaire bilingue: dessiner

Avec la bibliothèque turtle de Python.

mettre le stylo en position d'écriture

`pendown()`

relever le stylo

`penup()`

aller à x:  y: 

`goto(x,y)`

...

Dictionnaire bilingue: dessiner

Avec la bibliothèque turtle de Python.

mettre le stylo en position d'écriture

`pendown()`

relever le stylo

`penup()`

aller à x:  y: 

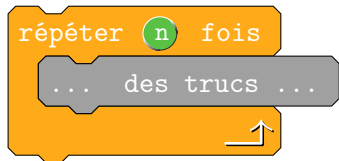
`goto(x,y)`

...

Un outil de transition?

carres

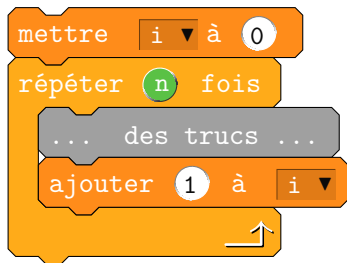
Dictionnaire bilingue: les boucles



```
for i in range(n):  
    # ... des trucs ...
```

Et si on veut accéder à l'indice de boucle en Scratch?

Dictionnaire bilingue: les boucles



```
for i in range(n):  
    # ... des trucs ...
```

iteration

robot (simple)

Boucles et variables

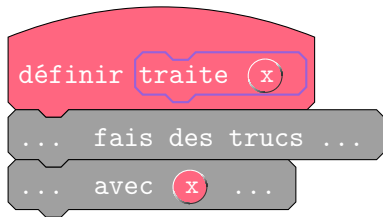
Le cas des boucles bornées est particulièrement délicat:

- ▶ Scratch est artificiellement bridé (pas d'accès au rang de l'itération).
- ▶ Python utilise un mécanisme d'itération très général
`for ... in ...: ...`
- ▶ en algo classique, on a tendance à beaucoup insister sur:
pour i allant de 1 à n
ce qui n'est naturel ni en Scratch ni en Python;
- ▶ *c'est un point explicite du programme dès le collège.*

Mais...

en Python	en Scratch
les valeurs ont un type unique (int, float, str, ...)	en Scratch les conversions sont implicites
on fait des boucles bornées sur tous les types séquen- tiels (et plus généralement les <i>itérables</i>)	la boucle bornée ne donne pas accès au rang d'itération

Dictionnaire bilingue: sous-programmes



```
def traite(x):  
    # ... fais des trucs ...  
    # ... avec x ...
```



```
traite(42)
```

frises

robot (avancé)

Similitudes

Scratch et Python sont tous les deux des langages:

- ▶ séquentiels;
- ▶ impératifs;
- ▶ interprétés.

En plus:

- ▶ la séquentialité = le passage à la ligne suivante;
- ▶ l'indentation en Python colle bien avec la présentation des sous-blocs en Scratch;
- ▶ les blocs personnalisés permettent d'introduire la notion de fonction.

Dictionnaire bilingue: fonctions

???

```
def f(x):  
    # ... fais des trucs ...  
    return une_expression
```

???

```
a=f(42)
```


Mais...

en Python	en Scratch
les valeurs ont un type unique (int, float, str, ...)	en Scratch les conversions sont implicites
on fait des boucles bornées sur tous les types séquentiels (et plus généralement les <i>itérables</i>)	la boucle bornée ne donne pas accès au rang d'itération
on évalue les fonctions dans les expressions	on exécute les blocs comme des instructions

Similitudes

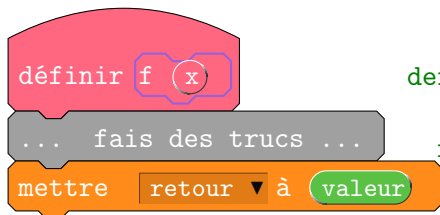
Scratch et Python sont tous les deux des langages:

- ▶ séquentiels;
- ▶ impératifs;
- ▶ interprétés.

En plus:

- ▶ la séquentialité = le passage à la ligne suivante;
- ▶ l'indentation en Python colle bien avec la présentation des sous-blocs en Scratch;
- ▶ les blocs personnalisés permettent d'introduire la notion de fonction `procédure`.

Dictionnaire bilingue: fonctions



```
def f(x):  
    # ... fais des trucs ...  
    return une_expression
```



```
a=f(42)
```

Pas très robuste...

Fonctions

C'est LA nouveauté majeure du programme de seconde:

- ▶ c'est une notion nouvelle et difficile;
- ▶ c'est la manière officiellement recommandée de présenter les algorithmes;
- ▶ le lien avec la notion de fonction en math est à exploiter et éclaircir;
- ▶ c'est le lieu où doivent se discuter des concepts comme la portée des variables ou l'ordre d'évaluation des expressions;
- ▶ en Python, une fonction est un objet comme les autres (on peut le passer en argument).

integrale

La programmation événementielle multi-agents

quand je commence comme un clone

créer un clone de

supprimer ce clone

quand je reçois ▼

quand ▼ est cliqué

c'est de la programmation orientée objet

c'est le travail d'une boucle d'évènements

La programmation événementielle multi-agents

quand je commence comme un clone

créer un clone de

supprimer ce clone

quand je reçois ▼

quand ▼ est cliqué

c'est de la programmation orientée objet

c'est le travail d'une boucle d'évènements

À quoi bon?

Similitudes

Scratch et Python sont tous les deux des langages:

- ▶ séquentiels (**pas Scratch**);
- ▶ impératifs;
- ▶ interprétés.

En plus:

- ▶ la séquentialité = le passage à la ligne suivante;
- ▶ l'indentation en Python colle bien avec la présentation des sous-blocs en Scratch;
- ▶ les blocs personnalisés permettent d'introduire la notion de ~~fonction~~ procédure.

Mais...

en Python	en Scratch
les valeurs ont un type unique (int, float, str, ...)	en Scratch les conversions sont implicites
on fait des boucles bornées sur tous les types séquentiels (et plus généralement les <i>itérables</i>)	la boucle bornée ne donne pas accès au rang d'itération
on évalue les fonctions dans les expressions	on exécute les blocs comme des instructions
on lance UN programme	les lutins suivent DES scripts en réagissant à des évènements (y compris des interactions entre lutins)

Pour conclure : l'écosystème

Une différence de cible:

- ▶ Scratch a été conçu *pour l'éducation*;
- ▶ Python est un langage généraliste.

Pour conclure : l'écosystème

Une différence de cible:

- ▶ Scratch a été conçu *pour l'éducation*;
- ▶ Python est un langage généraliste.

Donc:

- ▶ Scratch est prêt pour l'algo au collège;
- ▶ Python est prêt pour l'algo au lycée;

Pour conclure : l'écosystème

Une différence de cible:

- ▶ Scratch a été conçu *pour l'éducation*;
- ▶ Python est un langage généraliste.

Donc:

- ▶ Scratch est prêt pour l'algo au collège;
- ▶ Python est prêt pour l'algo au lycée;
- ▶ Python peut être étendu par des bibliothèques:
 - ▶ `turtle` pour la transition depuis le collège;
 - ▶ `math` pour les fonctions « scientifiques » usuelles;
 - ▶ `random` pour jouer avec l'aléatoire;
 - ▶ `matplotlib` pour le tracé de fonctions;
 - ▶ `sympy` pour le calcul symbolique;
 - ▶ *etc.*

C'est le dernier transparent

Si on arrive là, et qu'on a épuisé les discussions, deux amusettes :

marches