La transition Scratch/Python

Lionel Vaux Auclair

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

4e journée enseignement de l'info
3 avril 2019
Campus de Luminy, Marseille

Au collège

Le programme de cycle 4:

- `http://eduscol.education.fr/pid34185/cycle-4.html`

Au collège

Le programme de cycle 4:

- `http://eduscol.education.fr/pid34185/cycle-4.html`
- en maths (pp. 154/155), c'est succinct
- en techno (pp. 143/144) c'est ambitieux (et « boucle conditionnelle », c'est ambigu)

Au collège

Le programme de cycle 4:

- `http://eduscol.education.fr/pid34185/cycle-4.html`
- en maths (pp. 154/155), c'est succinct
- en techno (pp. 143/144) c'est ambitieux (et « boucle conditionnelle », c'est ambigu)

Les repères et attendus en math:

- `http://eduscol.education.fr/pid38235/5e.html` (en 5e), `http://eduscol.education.fr/pid38236/4e.html` (en 4e) et `http://eduscol.education.fr/pid38237/3e.html` (en 3e):

Au collège

Le programme de cycle 4:

- `http://eduscol.education.fr/pid34185/cycle-4.html`
- en maths (pp. 154/155), c'est succinct
- en techno (pp. 143/144) c'est ambitieux (et « boucle conditionnelle », c'est ambigu)

Les repères et attendus en math:

- `http://eduscol.education.fr/pid38235/5e.html` (en 5e), `http://eduscol.education.fr/pid38236/4e.html` (en 4e) et `http://eduscol.education.fr/pid38237/3e.html` (en 3e):
- c'est trois fois la même chose pour algo/prog, sauf le niveau d'attendus (1 à 3);
- au niveau 3, il faut savoir utiliser des boucles, accumuler des résultats dans des variables, utiliser les évènements, les blocs personnalisés, ...

Au lycée

Dans le cours de math de 2nde (lycée général):

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

Au lycée

Dans le cours de math de 2nde (lycée général):

- `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

Dans le cours de math de 2nde (lycée général):

- 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

Dans le cours de math de 2nde (lycée général):

- `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

Dans le cours de math de 2nde (lycée général):

- `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

Dans le cours de math de 2nde (lycée général):

- `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.

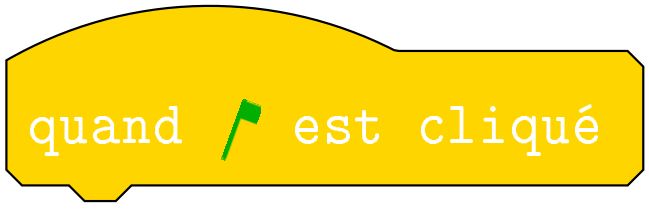
Projet de programme au LP: dans la même veine.

Similitudes

Scratch et Python sont tous les deux des langages:

- séquentiels;
- impératifs;
- interprétés.

Dictionnaire bilingue: démarrage



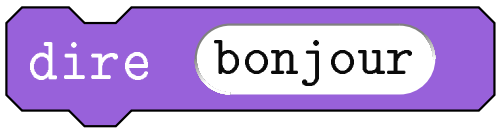
début du programme
(F5 dans IDLE, la flèche verte dans Pyscripter, l'appel à Python depuis la ligne de commande, . . .)

Dictionnaire bilingue: les variables

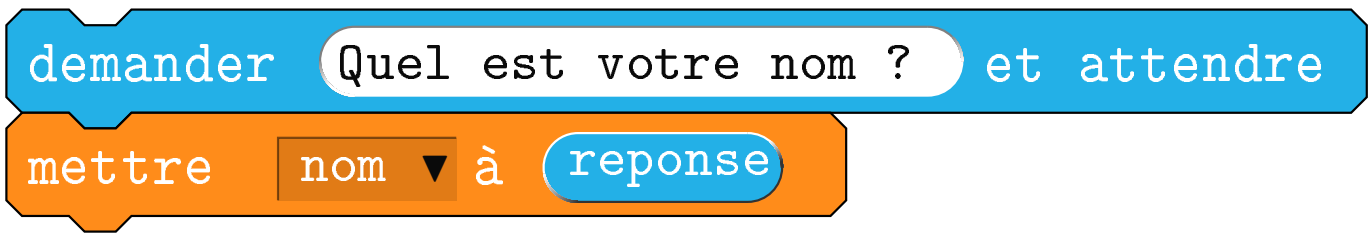

```
x = valeur
```


```
x = x+valeur
```

Dictionnaire bilingue: entrées/sorties




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




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

helloworld

Similitudes

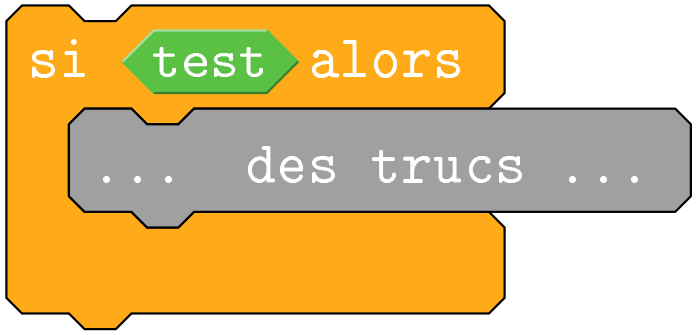
Scratch et Python sont tous les deux des langages:

- séquentiels;
- impératifs;
- interprétés.

En plus:

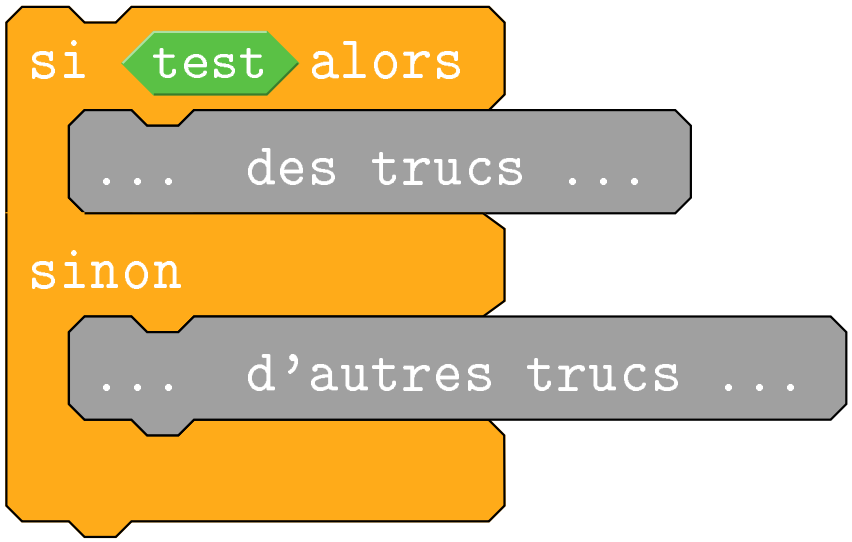
- la séquentialité = le passage à la ligne suivante;

Dictionnaire bilingue: les conditionnelles




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

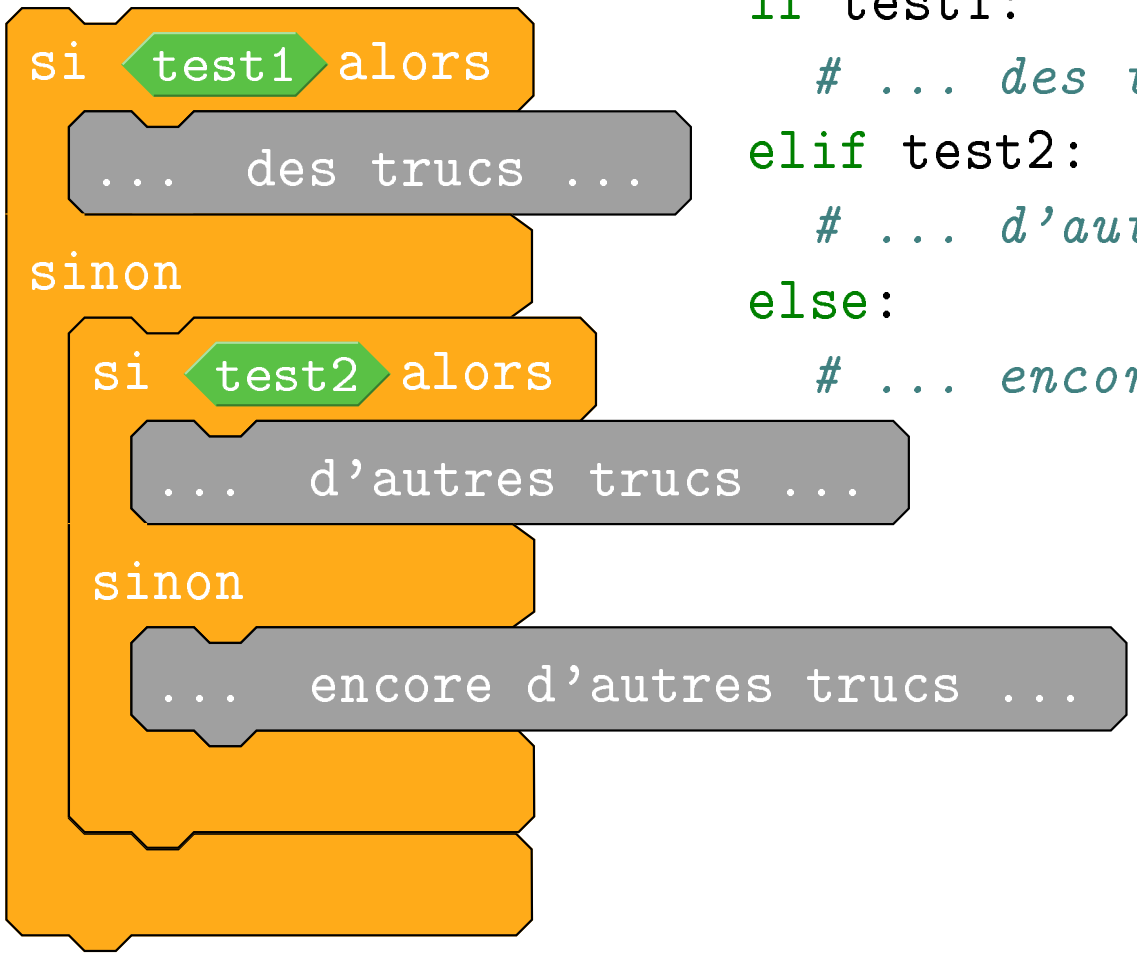
Dictionnaire bilingue: les conditionnelles




```
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.

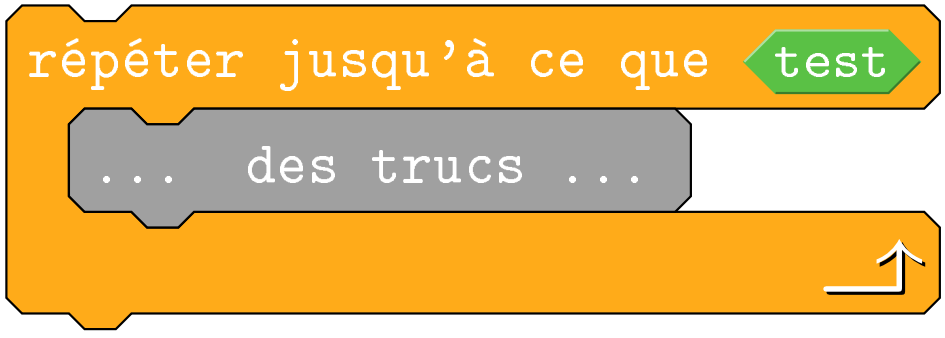
En plus:

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

Mais. . .

en Python	**en Scratch**
les valeurs ont un type unique (`int`, `float`, `str`, . . .)	en Scratch les conversions sont implicites

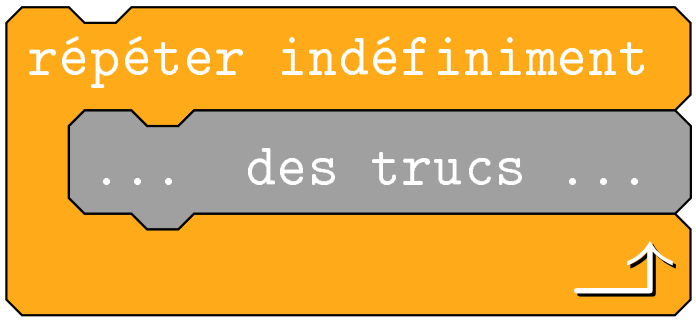
Dictionnaire bilingue: les boucles




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

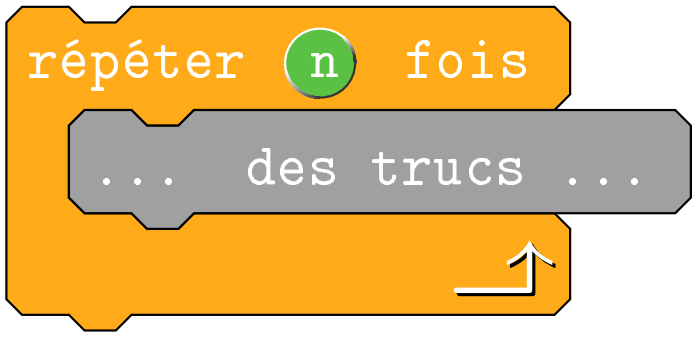
La condition est inversée!

Dictionnaire bilingue: les boucles




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

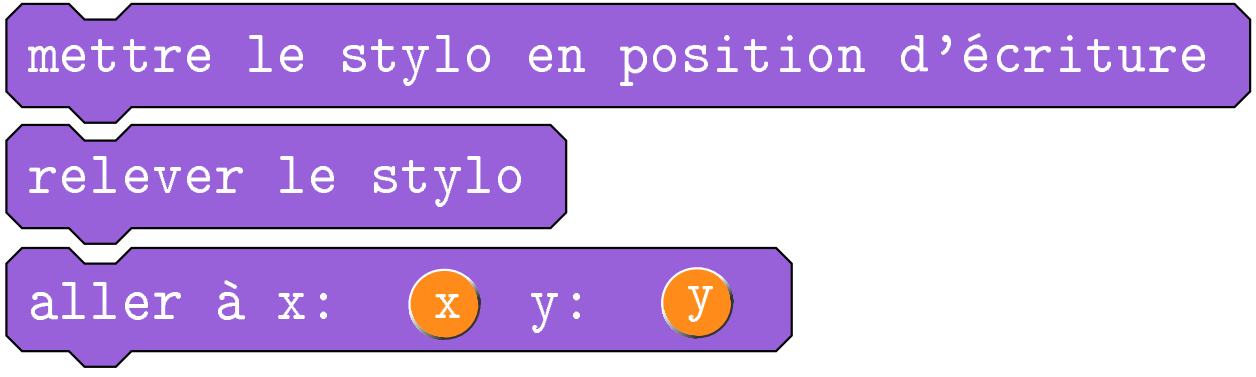
Dictionnaire bilingue: les boucles




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

Dictionnaire bilingue: dessiner

Avec la bibliothèque `turtle` de Python.

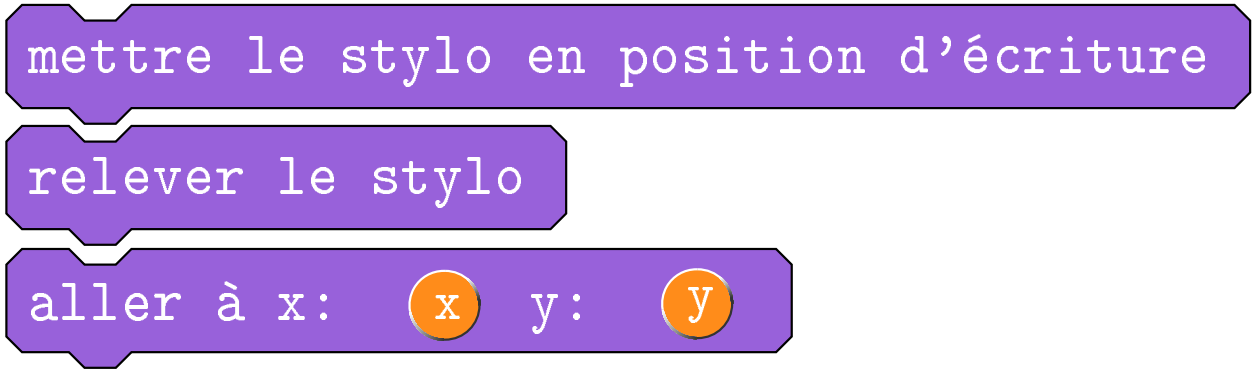



```
pendown()
penup()
goto(x,y)
```

...

Dictionnaire bilingue: dessiner

Avec la bibliothèque `turtle` de Python.



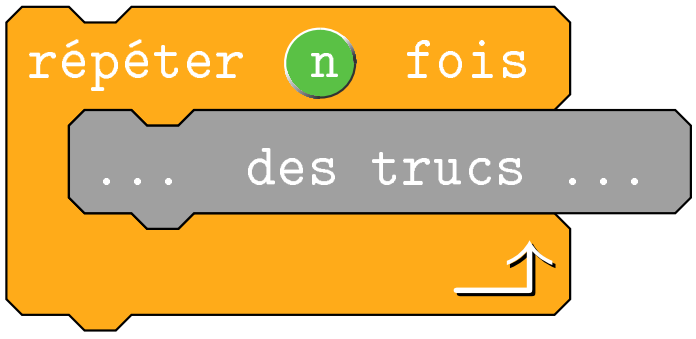

```
pendown()
penup()
goto(x,y)
```

...

Un outil de transition?

carres

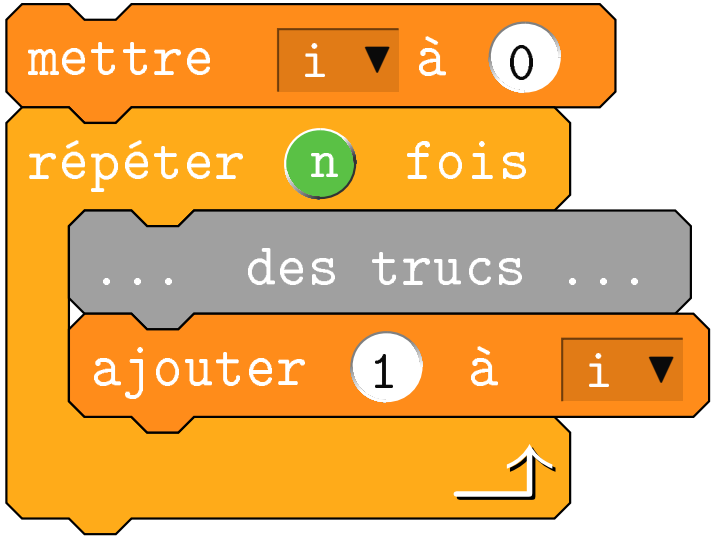
Dictionnaire bilingue: les boucles




```
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.*

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 *itérables*)	la boucle bornée ne donne pas accès au rang d'itération

Boucles et variables

Je cite (formation interne au lycée Diderot `https://pydiderot.readthedocs.io/formation/enseignants2/`):

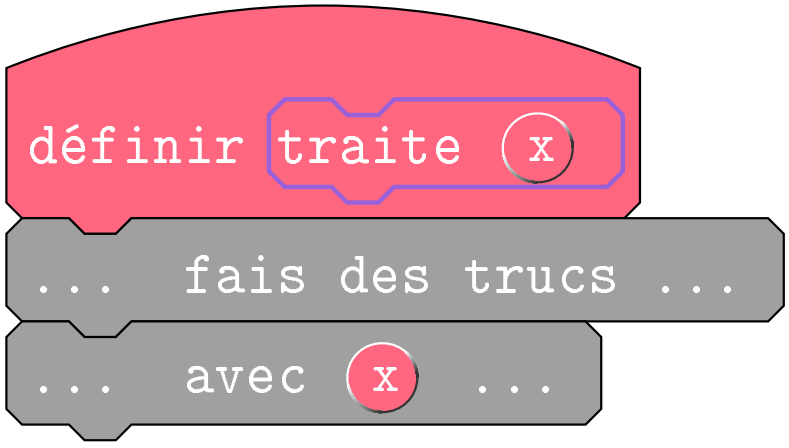
La notion de variable pose des problèmes aux élèves. Son introduction doit être considérée comme délicate [...]

Remarques générales : la notion de boucle pose moins de problème aux élèves que la notion de variable. On peut très bien faire des boucles sans variable, par exemple en blockly ou en « débranché ». L'instruction est alors du type « répéter 5 fois... ». Les élèves comprennent très bien l'intérêt de demander à l'ordinateur de répéter plusieurs fois les mêmes instructions. [...]

ce travail a un coût pour les élèves. Si on présente par exemple les quelques lignes de syntaxe permettant de faire une boucle « while », ces lignes ne paraissent pas naturelles. De plus, leur mémorisation par les élèves ne va pas du tout de soi. Enfin, ils ne comprennent pas forcément très vite à quel point ces quelques lignes permettent de résoudre un GRAND nombre de problèmes. Tous les ingrédients sont donc réunis pour que les élèves décrochent.

Propositions de remèdes : aller progressivement, faire prendre conscience de l'utilité de la structure de boucle while...

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 *itérables*)	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

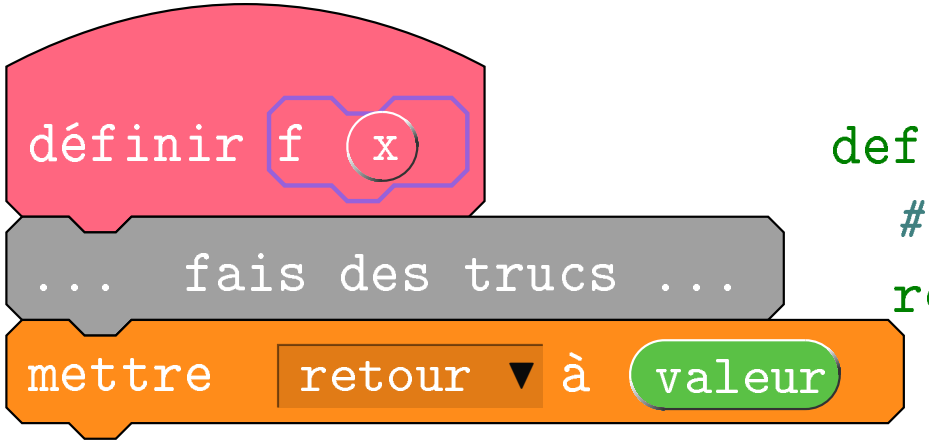
Scratch et Python sont tous les deux des langages:

- séquentiels;
- impératifs;
- interprétés.

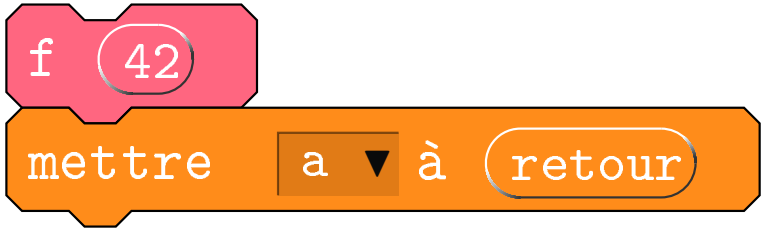
En plus:

- 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...

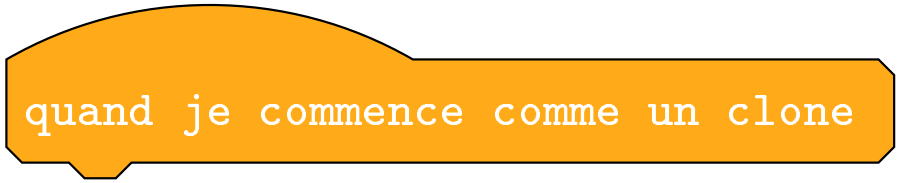
Fonctions

C'est LE point majeur du programme de seconde:

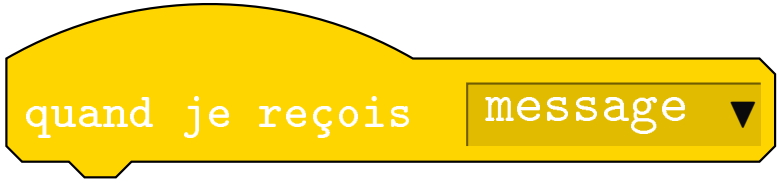
- c'est une notion nouvelle et difficile;
- c'est la manière officiellement recommandée de présenter les algorithmes;
- le lien avec la notion de fonction en math est à exploiter et éclaircir;
- c'est le lieu où doivent se discuter des concepts comme la portée des variables ou l'ordre d'évaluation des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

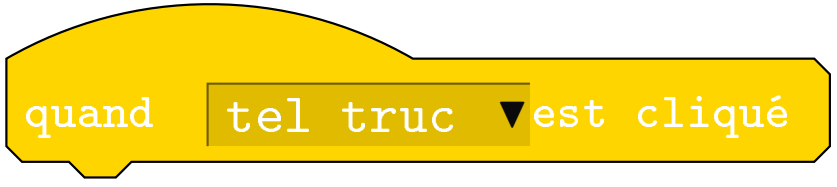
integrale

La programmation évènementielle multi-agents



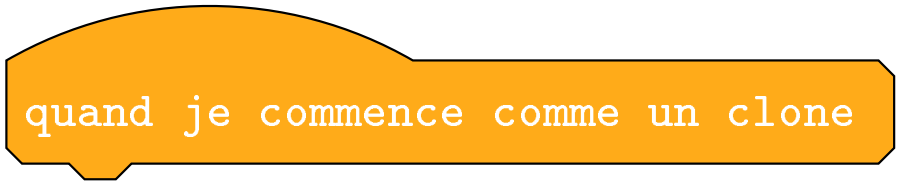
c'est de la programmation orientée objet



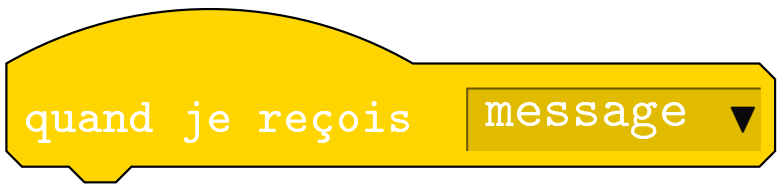


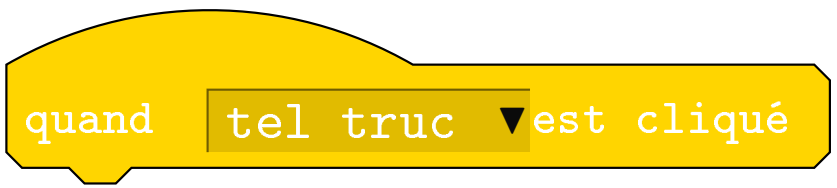
c'est le travail d'une boucle d'évènements

La programmation évènementielle multi-agents



c'est de la programmation orientée objet





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

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 *itérables*)	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)

L'écosystème

Une différence de cible:

- Scratch a été conçu *pour l'éducation* (cf. la conférence de ce matin, Papert, *etc.*);
- Python est un langage généraliste.

L'écosystème

Une différence de cible:

- Scratch a été conçu *pour l'éducation* (cf. la conférence de ce matin, Papert, *etc.*);
- Python est un langage généraliste.

Donc:

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

L'écosystème

Une différence de cible:

- Scratch a été conçu *pour l'éducation* (cf. la conférence de ce matin, Papert, *etc.*);
- 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;
  - `maplotlib` 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`