

Programmer avec *p5Visuel*¹ pour trouver les factorisations premières

Dans le film *Contact* de Robert Zemeckis, basé sur une nouvelle de Carl Sagan, voici comment on imagine notre premier contact avec une civilisation extraterrestre :



Voir <https://youtu.be/NLhw9POrpeE>

Ce qui est sous-entendu ici, c'est que les mathématiques constituent un langage vraiment universel, et que les nombres premiers en sont une composante inévitable.

La décomposition d'un nombre en facteurs premiers est un concept important en mathématiques. Mais pourquoi voudrait-on prendre la peine d'écrire un programme pour calculer cette factorisation ? Après tout, on peut trouver sur le web plusieurs pages faisant déjà cela pour nous². Mais ces pages web comportent peut-être des éléments non souhaités : publicité, limitations, écriture non-standard, etc. Et peut-être veut-on par la suite utiliser ce programme pour faire autre chose : par exemple, faire une liste des nombres naturels dans laquelle les nombres premiers sont soulignés, comme ci-contre où ils sont écrits en blanc sur fond rouge (voir le programme placé à l'adresse <http://profmth.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021b/factoPrem.html>).

Nombre	Factorisation
2	2
3	3
4	2 ²
5	5
6	2 × 3
7	7
8	2 ³
9	3 ²
10	2 × 5
11	11
12	2 ² × 3
13	13
14	2 × 7
15	3 × 5
16	2 ⁴
17	17
18	2 × 3 ²
19	19
...	...

¹ *p5Visuel* est un logiciel libre et gratuit disponible à l'adresse suivante

<http://profmth.uqam.ca/~boileau/p5VisuelWEB/index.html>

Dans la version web de l'article, tous les liens web de l'article seront actifs. Faire contrôle+clic (sur Windows ou Linux) ou commande+clic (sur Macintosh) pour ouvrir le lien dans une nouvelle fenêtre.

² Par exemple : <https://www.dcode.fr/decomposition-nombres-premiers>,
<https://www.deleze.name/marcel/culture/premiers/calculateur/index.html>
ou <https://www.mathepower.com/fr/facteurpremier.php>.

Il me semble cependant que la raison principale pour écrire un tel programme est la suivante : c'est une occasion pour approfondir à la fois ses connaissances mathématiques et informatiques.

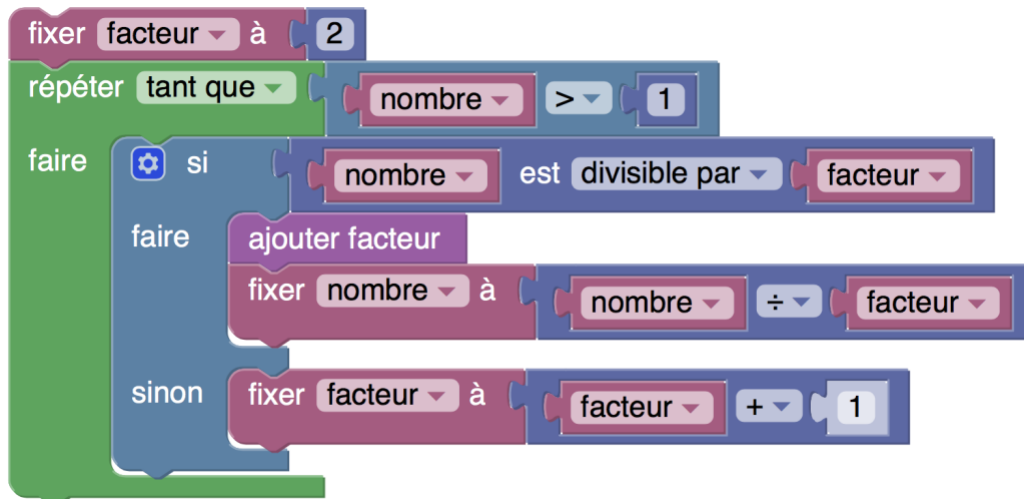
Voyons comment on peut procéder. Au départ, nous avons un *nombre* (qui est un entier strictement positif) et nous cherchons ses *facteurs*. Exception faite du facteur 1 qui ne nous intéresse pas, nous savons que ceux-ci se trouvent entre 2 et *nombre* inclusivement. Voilà comment nous allons procéder :

- au départ, nous posons *facteur* $\leftarrow 2$
- tant que *nombre* > 1
 - si *facteur* divise *nombre*
 - alors nous l'ajoutons aux facteurs trouvés
 - et nous poursuivons avec *nombre* $\leftarrow \text{nombre} \div \text{facteur}$
 - sinon nous poursuivons avec *facteur* $\leftarrow \text{facteur} + 1$

Voyons comment ça fonctionne avec un *nombre* valant 63 :

- comme 2 ne divise pas 63, on continue avec un *facteur* 3 (soit 2 + 1)
 - comme 3 divise 63, on garde le facteur 3 et le *nombre* devient 21 (soit $63 \div 3$)
 - comme 3 divise 21, on garde le facteur 3 et le *nombre* devient 7 (soit $21 \div 3$)
 - comme 3 ne divise pas 7, on continue avec un *facteur* 4 (soit 3 + 1)
 - comme 4 ne divise pas 7, on continue avec un *facteur* 5 (soit 4 + 1)
 - comme 5 ne divise pas 7, on continue avec un *facteur* 6 (soit 5 + 1)
 - comme 6 ne divise pas 7, on continue avec un *facteur* 7 (soit 6 + 1)
 - comme 7 divise 7, on garde le *facteur* 7 et le *nombre* devient 1 (soit $7 \div 7$)
 - on s'arrête car *nombre* n'est plus supérieur à 1
- Les facteurs conservés sont alors 3 (deux fois) et 7 : on a donc $63 = 3 \cdot 3 \cdot 7$.

Voyons maintenant comment écrire le fragment de programme correspondant avec *p5Visuel* :



Notons en passant que le bloc s'obtient en sélectionnant l'option correspondante du bloc de la catégorie **Math**.

Il faut aussi réfléchir à la manière dont on va présenter la factorisation. Une façon de procéder est la suivante :

- au départ, on crée une variable *factorisation* de type « texte » et initialement vide

```
fixer factorisation à ""
```

- à chaque fois que l'on trouve un *facteur*
 - * si *factorisation* est vide, alors $factorisation \leftarrow facteur$ (ceci ne survient donc qu'une fois)
 - * sinon alors on ajoute un signe de multiplication puis *facteur* au bout de *factorisation* (dans notre cas, le signe de multiplication sera « • »).

```

pour ajouter facteur
  si factorisation est vide
    faire fixer factorisation à facteur
  sinon
    fixer factorisation à créer un texte avec factorisation
    "•"
    facteur
  
```

Notons ici un fait remarquable : même si on n'exige rien des facteurs essayés, **tous** ceux-ci seront premiers. Pouvez-vous expliquer ceci? (Indice : les facteurs sont essayés en ordre croissant³.)

Pour terminer, on ajoute les instructions *p5Visuel* pour définir l'interface

```

pour actions initiales
  Marge gauche 40 pixels
  Créer un titre contenant "Factoriser le nombre ci-dessous"
  Créer une entrée entrée de nom "Nombre" et de contenu initial ""
  si on tape 'retour' dans entrée faire factoriser
  Créer un paragraphe réponse contenant ""
  
```

et pour obtenir la valeur entrée par l'utilisateur

```
fixer nombre à valeur de entrée
```

³ On utilise aussi la propriété mathématique suivante :
 si p divise $facteur$ et si $facteur$ divise $nombre$ alors p divise $nombre$.

On obtient alors le programme suivant

Factoriser le nombre ci-dessous

Nombre

$3 \cdot 3 \cdot 7$

Voir <http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021b/factoPrem1>

On peut aussi vouloir obtenir la factorisation première exprimée sous forme exponentielle, ce qui me donne l'occasion de souligner que *p5Visuel* rend facile l'utilisation de notations mathématiques

Factoriser le nombre ci-dessous

Nombre

$3^2 \cdot 7^1$

Voir <http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021b/factoPrem2>

Pour ce faire, il suffit de compter le nombre de fois qu'un même facteur apparaît

```
fixer multiplicité à 0
répéter tant que nombre est divisible par facteur
faire
  fixer multiplicité à multiplicité + 1
  fixer nombre à nombre ÷ facteur
ajouter facteur
```

puis de faire un affichage correspondant

```
pour ajouter facteur
  si factorisation est vide
  faire
    fixer factorisation à créer un texte avec facteur
    créer un texte avec "("
    créer un texte avec "^("
    créer un texte avec "multiplicité"
    créer un texte avec ")"
  sinon
    fixer factorisation à créer un texte avec factorisation
    créer un texte avec " • "
    créer un texte avec facteur
    créer un texte avec "^("
    créer un texte avec "multiplicité"
    créer un texte avec ")"
```

On termine en donnant à *p5Visuel* les instructions pour faire l’affichage du symbolisme mathématique

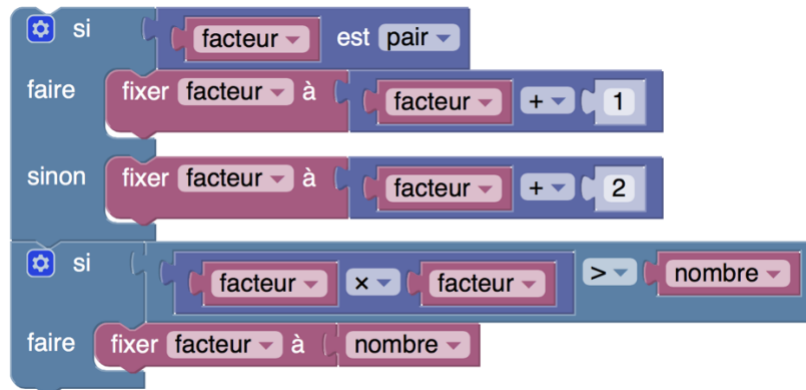


Défi : modifier le programme ci-dessus pour qu’il n’affiche pas les exposants égaux à un.

En terminant, on peut se demander s’il est vraiment nécessaire d’essayer un à un tous les facteurs. Un peu de réflexion mathématique nous permet de conclure qu’on peut remplacer la ligne



par les lignes suivantes



On obtient ainsi un programme un peu plus long, mais parfois beaucoup plus rapide

Factoriser le nombre ci-dessous

Nombre

$3^1 \bullet 62551441^1$

Voir <http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021b/factoPrem3>

Pouvez-vous expliquer ce qui se passe?

Un peu de recul

Nous venons de voir que nous pouvons exécuter une tâche de plusieurs façons, en utilisant des programmes différents. De plus, ces programmes peuvent avoir des caractéristiques spécifiques intéressantes : représentations différentes, rapidité améliorée, etc. Nous avons aussi constaté que les mathématiques peuvent nous aider, tant à vérifier que ces programmes fonctionnent correctement qu’à les rendre plus efficaces.

Enfin, même si plusieurs environnements de programmation peuvent être utilisés pour réaliser cette démarche (ne serait-ce qu’en partie), *p5Visuel* nous a permis de la réaliser complètement, simplement, et rapidement.