Donner du sens aux éléments de technologie : soulevons le capot du numérique

Pour citer cet article :

Alvarez Aurélien, Garreau Pascal, Tort Françoise et Viéville Thierry (2014). Donner du sens aux éléments de technologie : soulevons le capot du numérique. Deuxième volet d'une série de deux articles. *Adjectif.net*, [En ligne] http://www.adjectif.net/spip/spip.php?article294

Résumé :

Le BO°8, 13/11/2011 de l'enseignement de spécialité Informatique et Sciences du Numérique (ISN) précise la finalité de cet enseignement « maîtriser les mécanismes fondamentaux qui régissent ces mutations [du numérique] et être en mesure d'apprécier les enjeux sociétaux qui en découlent ». Un élément clé est de *donner du sens aux éléments de technologie et à leurs usages*. Notre thèse va être ici de montrer qu'apprendre à « coder » (d'aucuns diront programmer) n'est qu'un marchepied pour apprendre à *décoder* le numérique, du plus petit à la plus grande d'entre nous.

Mots clés :

Apprentissage de l'informatique, enseignement de spécialité Informatique et Sciences du Numérique (ISN)

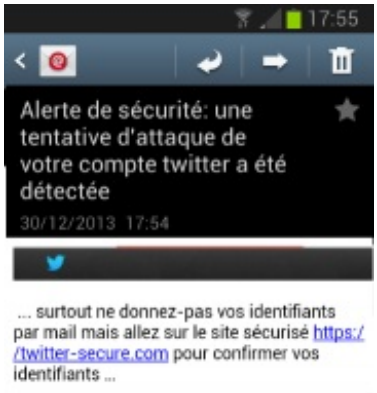
Questions de sécurité

Quand l'adolescence est là : « Internet on connait, ça va ». On sait [1] « mieux que les vieux comment ne pas se taper la honte sur les réseaux sociaux ». Et parfois, « beurk les cours de techno ».

Ce qu'il nous faudrait, c'est un cas concret et parlant où expliquer une technologie permet de résoudre certains problèmes d'usage que nous pourrions rencontrer. D'ailleurs … *oh* …

Excusez-moi, mais, voilà que je viens de recevoir un courriel inquiétant.

C'est un mél qui semble sérieux : pas de *fôte* d'orthographe et une invitation à aller sur le site sécurisé mentionné, qui affiche cette page :

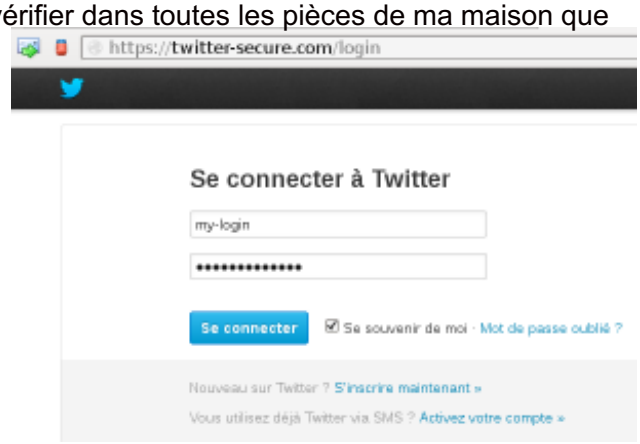


C'est exactement la bonne présentation de Twitter, il y a bien « twitter » dans l'adresse et en plus je crois me souvenir que le « s » de « https: » veut dire sécurisé. Alors ?

Alors ? Me voilà à un clic de me faire bêtement « harponner » mes identifiants Twitter. Parce qu'il n'y a aucune raison que twitter-secure.com soit un site de Twitter.

C'est le coup classique à double bande : un voleur tente de me cambrioler mais fuit rapidement devant les uniformes de deux policiers qui passaient devant ma

maison ; ces deux hommes ont même la gentillesse de venir vérifier dans toutes les pièces de ma maison que rien ne manque. Et de repartir avec mes biens les plus précieux... Voleurs de connivence avec le premier, sous de faux uniformes, pour endormir ma méfiance.



On a bien compris l'arnaque. Certes. Mais, il est difficile d'envisager tous les cas possibles de tentatives de harponnage passées et surtout futures qui seront basées sur une manipulation de « l'adresse Internet » qui n'est pas encore inventée !

Il y a une parade universelle : apprendre une bonne fois pour toute *la langue dans laquelle s'écrivent ces adresses*. Dans ce cas, nous n'aurions plus à « ré »-apprendre chaque usage, car il ne sera qu'une conséquence de la culture scientifique et technique que nous maîtriserions. Ce que nous rapportons ici n'est pas un à priori, mais bien ce que l'on peut constater lors des actions de médiation scientifique sur ces sujets.

De la notion d'identificateur de ressources universel

Très simplement, il s'agit d'apprendre à chacune et chacun ce qu'est un Identificateur de Ressource Universel ou une *URI*, un objet informatique bien spécifié qui permet d'identifier toutes les ressources numériques, de manière unique et de pouvoir aussi (i) *y accéder* et (ii) *les utiliser*. Cet apprentissage répond donc aux deux questions concrètes suivantes :

> - *Comment accéder à cet objet numérique ?*
> - *Comment le faire fonctionner ?*

Les exemples proposés ici illustrent tous ces éléments et, lorsque l'on soumet cette figure à un auditoire non averti, les personnes devinent très rapidement ou sont rapidement convaincues des éléments suivants :

- il y a des identificateurs qui permettent d'accéder à des pages Web (ceux qui commencent par « http: » ou « https: ») et d'autres qui permettent d'accéder à d'autres ressources Internet (par exemple, le mail avec ce qui commence par « mailto: » ou peut-être le chauffage de la maison s'il est connecté).

Nous voilà en train de faire comprendre la différence entre le réseau des réseaux, Internet, et une de ses applications, le Web.

- Ceux qui commencent par « file : » doivent concerner des fichiers locaux à l'ordinateur au contraire de ressources qui sont distantes.

Nous voilà en train de faire comprendre un lien entre la localisation physique et logique des ressources.

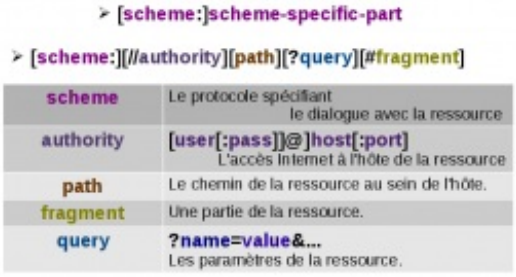
- Rappelons que l'« ISBN » (pour International Standard Book Number) désigne de manière unique un livre. Cela permet de mesurer que la notion de URI concerne donc à la fois :
  - la localisation de ressources (on parle de URL, L pour location) et
  - le nom (non ambigu) de ressources (on parle de URN, N pour *name*).
- Ces URI ne se présentent pas forcément sous forme textuelle :
  - les QR-code permettent de les encoder sous forme d'image et de les faire relire par la machine sans les retaper,
  - les menus de configuration permettent souvent d'entrer les éléments d'une URI.
- On peut passer des « paramètres » à une URI, par exemple :
  - celle avec « ceparou06.fr » semble (à qui sait que c'est « ceparou06 » renvoie au site de transports publics du département 06) donner un horaire précis,
  - « zimbra.inria.fr » semble (à qui sait que « zimbra » est un logiciel d'agenda) [presque] donner l'emploi du temps du PDG Inria,
  - « wiki.inria.fr » semble (à qui sait qu'un wiki est un site d'édition de texte) donner la page en mode imprimable,
  - « mailto: » semble permettre d'écrire un courriel avec un sujet et un destinataire donné.

En observant tous les exemples donnés ci-dessus, nous voyons la richesse explicative qui est offerte par cette exploration : elle permet de donner le sentiment objectif que comprendre cet objet et éclaire simultanément de multiples pratiques numériques. Ce qui rend le partage de savoir attrayant, c'est que chaque brique d'explication, conduit à éclairer un « petit grain » du quotidien numérique.

Comment lire un URI ?

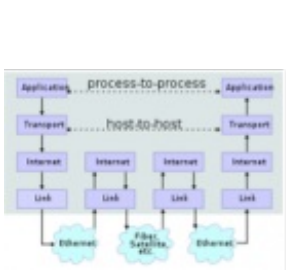
Le langage des URI est un petit langage informatique, une convention universelle pour désigner une entité. Ce sont des phrases comptant, au plus, cinq mots :

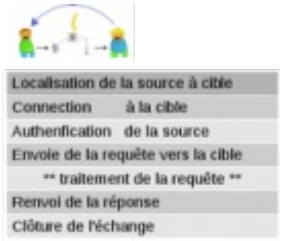
Ici les symboles '//', '/', ' ?', '#' sont des préfixes qui permettent de déterminer quel mot de la phrase va suivre.



Le « schème » définit le *protocole à utiliser pour accéder à la ressource*. Cela peut-être « http: » ou « https: » pour accéder à une page Web, « file : » pour accéder à un fichier de l'ordinateur, « mailto: » pour créer un courriel. On peut aussi définir des protocole *ad hoc*, par exemple « dev : » pour accéder à un dispositif (un *device*) relié à l'ordinateur. C'est alors le moment d'expliquer comment Internet est architecturé et la raison d'être des protocoles, au niveau de détail schématisé ici :

Cela permet aux mécanismes de prendre du sens et d'éclairer les usages que nous faisons de ces objets numériques.





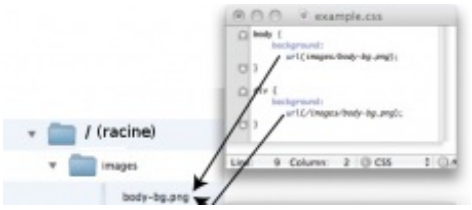
L'« *authority* » définit l'accès à la *machine où est la ressource*. On est en train d'évoquer la couche de « transport » du réseau. Cet accès sera entièrement défini par :

- le *nom de domaine* de la machine (par exemple

« google.com » ou « zimbra.inria.fr ») ou son *adresse IP* (par exemple « 66.249.91.99 »), voire le numéro de port de connexion à la machine ;

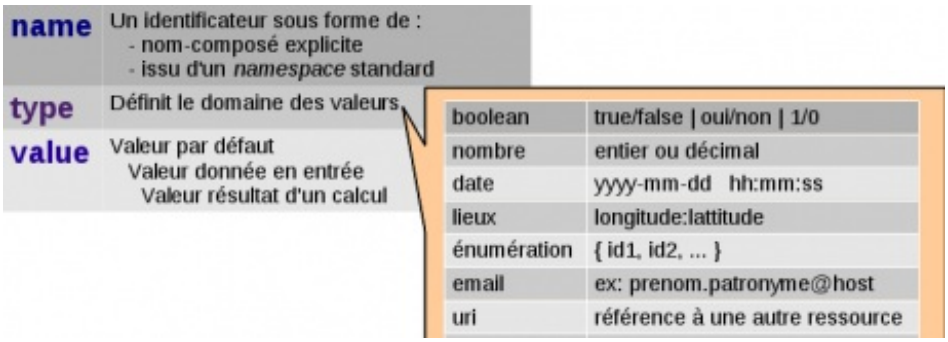
- complété des identifiants de connexion éventuels (par exemple « monidentifiant:môdepasse@zimbra.inria.fr » ou, si le mot de passe n'est pas utile, « monidentifiant@zimbra.inria.fr »).
- Le « *path* » : une fois connecté à la machine, pour trouver la ressource, il est naturel de spécifier son chemin. C'est une construction de la forme : « dossier/sous-dossier/nom-du-fichier.extension-du-fichier »

C'est surtout l'occasion de vérifier que nous comprenons bien :



- la structure arborescente d'un système de fichiers ;
- que l'extension d'un nom de fichier définit son « type » (texte simple, document textuel, image, son, logiciel, …) et donc la façon de l'*ouvrir* ou de l'*éditer* ;
- que le « *fragment* » va permettre d'accéder à un fragment de la ressource (par exemple « page=13 » pour un document paginé ou la division d'un texte qui a été marqué d'une « ancre » de la forme « » pour se positionner de manière unique dans le contenu) ;
- que la « *query* » va permettre de spécifier des paramètres de la ressource. Elle est de la forme : _ ?name=value&... où '=' et '&' sont des caractères qui permettent de délimiter le nom des paramètres de leur valeurs respectives et où '…' signifie que l'on peut passer 1, 2 ou plusieurs valeurs de paramètres. Ce dernier ingrédient permet d'introduire la notion de « t-uples » où, pour décrire l'état d'un système, on donne une liste de paramètres dont les valeurs correspondent à un type donné.

  Ce qu'il faut savoir se résume dans le diagramme ci-dessous :



  Des exemples de types de données courantes permettent de comprendre le besoin de bien définir de quoi on parle. On note aussi, pour faciliter l'usage de ces paramètres, quelques bonnes pratiques, comme :

  - utiliser des termes explicites et standards pour nommer les paramètres,
  - toujours prévoir une valeur par défaut des paramètres non spécifiés.

À quoi bon apprendre en détail à lire un URI ?

Apprendre à lire un URI signifie mettre du sens sur les éléments précédents pour éclairer des éléments que nous utilisons au quotidien. Pour revenir à l'exemple introductif, savoir lire un URI permet de se prémunir contre une méthode pernicieuse pour tromper un internaute : l'attaque de l'homme du milieu. Si on va sur le site « www.labanquepostal.fr » on ne voit pas qu'il manque le « -

e » à « postale » : la page est évidemment jumelle de la vraie ... pour mieux nous tromper ! Maintenant nous savons où se trouve le nom de la machine à laquelle on se connecte, donc *quelle partie* regarder attentivement, nous sommes capables de déterminer s'il y a arnaque !

Analyser un URI est aussi un levier pour présenter différentes couches du réseau (en l'occurrence les couches de transport à applicative). Se demander comment est lu l'URI sur Internet permet de décrire les différentes étapes d'un échange (connexion, authentification, etc..).

Pour les plus avides de théorie et au-delà de notre propos, analyser la syntaxe d'un URI ouvre la porte à l'étude des expressions régulières : cela montre que l'on peut apprendre une notion informatique comme celle-là, à un niveau très simple (juste comprendre les *wildcards* des fichiers) et opérationnel (faire des substitutions simples dans des chaînes), ou plus évolué : y compris ces liens avec l'algèbre des monoïdes, pour les plus matophiles.

Voici donc la thèse défendue :

- apprendre des rudiments d'informatique facilite fortement l'instrumentation des machines traitant des informations numérisées ;
- il y a relativement peu de notions à apprendre (elles sont résumées en moins de trois pages ici) ;
- elles peuvent s'apprendre par « l'exemple » en déconstruisant des objets numériques du quotidien ;
- chaque savoir donne un savoir-faire utile pour manipuler les objets numériques ;
- cet ensemble de connaissances et de compétences permet un savoir-être approprié, y compris face aux futurs usages du numérique ;
- décortiquer les URI semble l'un des leviers possibles pour lier usages et fondements théoriques,
- et permet d'ouvrir la discussion vers l'idée d'un référentiel de savoirs culturels à partager.

Nous proposons ici une ébauche d'identification des aspects importants de l'enseignement d'informatique dont un-e citoyen-ne a besoin, à partir d'un « truc » qu'on utilise tous les jours. Le « piger » permet de maîtriser des pans entiers des abstractions informatiques à comprendre pour appréhender les objets informatisés.

Ce *truc*, URI, qui est sous notre nez à longueur de navigation d'Internet *prend alors du sens* n'est pas un simple identifiant opaque : l'exemple de son décodage justifie les démarches visant l'appropriation des fondements de l'informatique au-delà de ses usages, afin de maîtriser, et non pas uniquement consommer, ce qu'Internet peut nous proposer.

Apprendre à écrire un URI ?

Bien entendu, l'étape suivante est d'apprendre à écrire des URI (par exemple pour accéder à des sites dynamiques). Deux activités peuvent être assez faciles à mettre en œuvre :

- aller chercher un contenu sur wikipédia pour alimenter sa propre page avec la bonne URI : cela permet de démystifier la notion de Web service et d'entrevoir la motivation du Web 3.0 (avec Dbpedia dans ce cas) ;
- imaginer la spécification d'un URI pour piloter un réfrigérateur domotisé imaginaire, y compris la documentation pour permettre de comprendre les fonctionnalités proposées, avec des bonnes pratiques de génie logiciel.

Si on apprend à coder, alors la couche des sockets TCP est facilement programmable (en Java, Python, …) pour relier ces notions à leur implémentation en termes de composants logiciels. Mais

cela nous emmène sûrement au niveau du lycée. Nous y arrivions justement.

Peut-on apprendre à écrire d'autres objets numériques ?

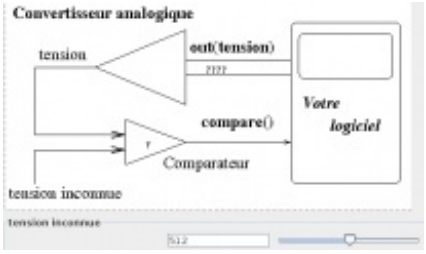
Les objets numériques sont le quotidien des jeunes : ces applications des smartphones ; ces documents pluri-média qui associent texte, son et images ou animations ; ces objets 3D virtuels qui réagissent quand on les manipule. Comment se donner les moyens de pas uniquement les utiliser mais **de soulever le capot**, de les paramétrer, programmer, changer ? De faire comme Barak Obama le disait dans son allocution à propos de « one hour of code » : « ne pas uniquement jouer à un jeu vidéo, mais aussi créer le sien ».

Les plateformes qui permettent de s'initier au codage se multiplient, de http://scratch.mit.edu à http://www.codecademy.com, mais elles se limitent finalement à apprendre à « coder », c'est-à-dire à découvrir les rudiments de la programmation [2]. Mais à quoi bon en soi ? Si ce n'est mieux comprendre *le numérique*.

Une notion permet de mettre en place de manière relativement optimale une activité de ce type, c'est celle de « proglet ». Une « proglet » est une petite applette [3] qui permet de s'initier à la programmation de manière ludique. Elle se présente sous forme d'une interface mettant en scène le/les concept(s) clé(s) à manipuler, sous forme visuelle et/ou sonore, à la fois pour mieux comprendre l'objet numérique et pour offrir une démarche expérimentale concrète. Une « proglet » est donc un objet numérique que l'on manipule en le programmant (en non en cliquant).

Des exemples de proglets

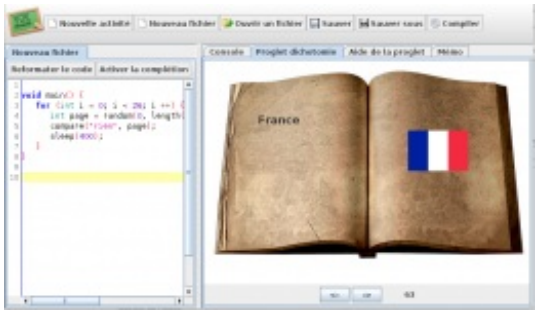
Prenons le principe algorithmique abstrait probablement le plus élémentaire : le mécanisme de dichotomie (on ne le rappelle pas ici). Il peut se décliner, par exemple :



- pour trouver le zéro d'une fonction réelle continue monotone dans un intervalle,
- pour réaliser un convertisseur analogique-digital,
- ou pour trouver un mot dans un dictionnaire. On a donc cité là trois contextes très différents pour un même principe algorithmique.

Des *proglets* permettent d'expérimenter ce principe unique dans les contextes cités. Pour cela, un panneau graphique minimal est programmé. Par exemple, à droite, la *proglet convertisseur* est un dessin qui montre sur le schéma électrique les deux fonctions disponibles : « out(double valeur) » pour sortir une valeur analogique à comparer et « int compare() » qui lit la valeur 1 ou -1 sur le comparateur. Sur ce schéma, les valeurs électriques s'affichent au fil du programme de l'apprenant. Bien entendu, c'est une simulation et la tension électrique est choisie à la main en bas du panneau graphique, pour tester le programme dont le déroulement sera observable.

À gauche, l'environnement complet d'une autre *proglet* est montré, mettant en œuvre un graphisme où on « tourne » les pages d'un livre, pour y rechercher un pays : panneau graphique, éditeur de code, document sur la *proglet* et texte de l'activité sont présentés dans le même environnement, c'est une *jarre Java* qui se lance sur un simple clic Web sur toutes les machines, rendant immédiat le démarrage et la prise en main de l'activité.

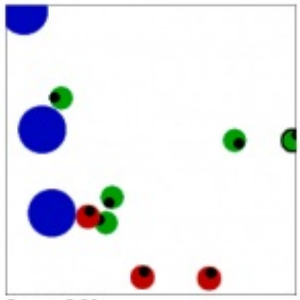


On a ainsi atteint les objectifs fixés : environnement convivial, ergonomie naturelle, documentation intégrée. Il n'y a plus qu'à se concentrer sur l'essentiel : le savoir et la pédagogie.

Proglet et initiation aux médias

Une plateforme telle que javascool.fr a permis d'expérimenter à grande échelle ce mécanisme didactique. On parle d'une vingtaine de *proglets* [4] utilisées par plusieurs centaines de classes de lycée, dans un cadre scolaire.

L'idéal serait d'utiliser ce même paradigme en tant que composant d'une page Web. L'apprenant est donc devant une activité de création de contenu pluri-média, qui comporte des éléments logiciels, là encore proposés packagés avec une activité à dérouler pour programmer la mise en fonctionnement des ingrédients proposés.



Sous le terme de *carpasinivore*, un contenu expérimental a été créé où de petits êtres minimalistes sont dans un environnement de survie et l'apprenant, au lieu de jouer en cliquant, doit programmer un « bot » : une fois lancé dans le jeu, ce « bot » ne doit sa survie qu'à la qualité de son algorithme.

La sur-simplification permet à la fois l'exécution raisonnable du *grain*, sa prise en main rapide, mais surtout ... **de soulever le capot !** Une fois l'activité achevée et complètement soldée, l'apprenant va naturellement se dire : « ah tiens, mais comment il a été fait ce grain ? » ou « comment pourrais-je le démonter / changer / etc. ? ». Bonne nouvelle, le *grain* est simple, accessible et prêt à être « détourné ».

L'utilisation de la plateforme github se prête naturellement à la création de ces branches, où chacune et chacun peut créer ses propres variantes.

On crée donc un chemin entre des activités complètement encadrées où il faut copier/coller des bouts de codes pour en observer le fonctionnement, puis remplir des codes « à trou », implémenter un principe algorithmique bien décrit, concevoir l'algorithme jusqu'à ... créer son propre objet numérique.

Conclusion : « dis mon enfant, que vas-tu faire du monde numérique qui est tien ? »

Les jeunes d'aujourd'hui feront le Web de demain. La tendance montre qu'en tant que simples internautes - sans parler de leurs besoins professionnels - , ils seront de plus en plus susceptibles de naviguer entre posture de consommateurs et de créateurs de contenus et services. Donnons-leur donc les bases algorithmiques mais aussi éthiques, légales, culturelles, économiques - et pourquoi pas esthétiques ! - pour qu'ils imaginent l'Internet de demain en toute connaissances de cause.

Pour citer cet article :

A. Alvarez, **P. Garreau,** F. Tort, , **T. Viéville,** «

Donner du sens aux éléments de technologie : soulevons le capot du numérique. », *Adjectif* [En ligne], mis en ligne le Lundi 9 juin 2014. URL : http://www.adjectif.net/spip/spip.php?article294

Haut de page

[1] « y'a plus risqué que Facebook pour mourir de honte, c'est la cour du collège et ça fait des siècles qu'on y traine » (citation authentique, débat public "science et société" entre chercheurs en informatique et jeunes de quartier).

[2] Par *codage*, comme dans la formule « apprendre à coder pour décoder le numérique », on entend dans les

faits « s'initier, par le savoir-faire, aux rudiments minimaux de la programmation (plus précisément aux quatre structures de contrôle : séquence, tests, affectations et boucles du paradigme de programmation impérative, complétés de quelques méthodes d'encapsulation par des fonctions ou des objets informatiques) de façon à pouvoir créer des objets numériques dotés de mécanismes algorithmiques. C'est un apprentissage qui prend environ une demi-journée avec les outils didactiques modernes. C'est une démarche intéressante qui se popularise largement et qui a plusieurs effets vertueux : apprentissage par le savoir-faire, mise en position d'acteur et de créateur de l'apprenant, démystification de ce qu'est l'informatique.

[3] ou "un logiciel qui s'exécute dans la fenêtre d'un navigateur web", issu de la page Wikipédia

[4] pour les algorithmes mathématiques, la manipulation d'images, de sons, de graphes, la simulation de la bonne vieille tortue logo, des activités autour d'une interface série, du cryptage RSA, du dessin pixelique, etc.

A. Alvarez

Articles de cet auteur

- Donner du sens aux éléments de technologie : jouons avec nos enfants
- [...]

F. Tort

Articles de cet auteur

- Donner du sens aux éléments de technologie : jouons avec nos enfants
- [...]

P. Garreau

Articles de cet auteur

- Donner du sens aux éléments de technologie : jouons avec nos enfants
- [...]

T. Vieville

Articles de cet auteur

- Donner du sens aux éléments de technologie : jouons avec nos enfants
- [...]