



ACADÉMIE DE NANTES

*Liberté
Égalité
Fraternité*

*

GRAFF SNT-NSI 2022-2023

BARRAUD Mickaël
CACCIOPPOLI Sandra
MARTINEAU-ROBIN Ursula

*

Socle de connaissance en SNT pour la NSI

Réflexions autour de la constitution et de la mise en œuvre d'un socle de connaissances à apporter aux élèves de seconde en SNT, en vue d'un bon démarrage en NSI. Acquisition d'automatismes via des traces écrites sur un carnet de savoirs et savoir-faire et, en programmation, par la répétition et la diversité des exercices.

Contenu de la ressource

1 Analyse	page	1
1. Axes de travail en SNT		1
2. Ciblage des éléments du BO		1
2 Travail sur la mise en œuvre	page	3
1. Les exercices		3
2. Le cours		7
3. Propositions d'une mise en œuvre		8
3 Exemples concrets	page	9
1. Programmation		9
2. Internet		10
3. Web		10
4. Réseaux sociaux		11
5. Les données		11
6. Liste de questions à mettre en forme (+répondre + à retenir)		11

1. Axes de travail en SNT

Les axes de travail

Il y a, depuis le début de l'introduction de l'informatique dans l'enseignement, deux axes de travail qui ont plus ou moins bien cohabité. L'utilisation des outils ou l'enseignement de la science informatique. Plus récemment, et notamment avec la SNT, s'est ajouté un axe sociétal.

★ L'informatique « outil »

On peut citer l'utilisation d'un traitement de texte, d'un tableau, ou de n'importe quel logiciel d'acquisition, d'exploitation, ou de traitement de donnée...

★ L'informatique « science »

On peut citer l'apprentissage des algorithmes et de la programmation ; la conception des réseaux ou des bases de données ; La compréhension des structures de donnée...

★ L'informatique « sociétal »

On peut citer la connaissance de la loi, le RGPD, les droits et licences, ou le problème du cyber-harcèlement. Mais aussi l'impact des objets connectés ou de l'abondance des données...

La science informatique

Les différents axes décrits ci-dessus sont tous à considérer pour l'enseignement en SNT. On prend cependant, ici, le point de vue de l'enseignant de NSI qui se dit parfois

« Si tous mes élèves avaient vu/maitrisé ça l'année dernière, ce serait plus facile... »

Le propos de la partie suivante est d'essayer de déterminer le « ça » en question à travers la lecture des programmes officiels. Mais d'expérience en première NSI et à la vue des attentes de Baccalauréat en Terminale NSI, une chose est sûre : cela tourne très clairement autour de l'axe : **Science informatique**.

2. Ciblage des éléments du BO

Il s'agit ici de tenter de dresser une liste des capacités attendues des élèves en SNT répondant aux critères suivants :

- ★ Cela relève de la science informatique
- ★ C'est une base utile pour le cours de spécialité NSI
- ★ C'est bien une base et non une notion avancée, qui sera, de toute façon, entièrement retravaillé en NSI

Autrement dit : on cherche les pré-requis pour la NSI atteignable en SNT :

Notions transversales de programmation

Contenus	Capacités attendues
Affectations, variables Séquences Instructions conditionnelles Boucles bornées et non bornées Définitions et appels de fonctions	Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes physiques, économiques et sociaux.

« Définitions et appels de fonctions » est un contenu difficile et qui demande un plus long temps d'appropriation. C'est pourquoi on ne l'a pas retenu pour la suite. Il nous semble que les fondamentaux sont **L'affectation**, les **conditionnelles** et les **boucles**.

Internet

Contenus	Capacités attendues
Protocole TCP/IP : paquets, routage des paquets	Distinguer le rôle des protocoles IP et TCP. Caractériser les principes du routage et ses limites. Distinguer la fiabilité de transmission et l'absence de garantie temporelle.

Sur ce sujet, le programme de première NSI demande explicitement : « Le lien est fait avec ce qui a été vu en classe de seconde sur le **protocole TCP/IP** ». Il semble donc bien un incontournable.

Le Web

Contenus	Capacités attendues
Hypertexte	Maîtriser les renvois d'un texte à différents contenus.
Langages HTML et CSS	Distinguer ce qui relève du contenu d'une page et de son style de présentation. Étudier et modifier une page HTML simple.
URL	Décomposer l'URL d'une page. Reconnaître les pages sécurisées.

L'**hypertexte** est une notion clé ici, qu'il nous semble devoir être travaillé aussi bien indépendamment en débanché qu'à travers le langage du Web. Il nous semble important, à ce niveau, que la distinction soit explicite entre le Web et Internet.

Les **langages HTML et CSS** devraient être abordés plus en profondeur que de « distinguer » ou « modifier ». En première NSI, c'est l'apport des interactions avec l'utilisateur qui est à travailler : cela sous-entend que ces langages ne soient pas un obstacle.

Concernant les **L'URL**, il serait pertinent que les élèves sachent distinguer l'adressage sur internet ou en local ; relatif ou absolue.

Les réseaux sociaux

Contenus	Capacités attendues
Rayon, diamètre et centre d'un graphe	Déterminer ces caractéristiques sur des graphes simples.

C'est ici le vocabulaire et la familiarisation avec les **graphes** qui est important, plus que les techniques de calcul. Le travail sur les graphes n'est un objectif de la spécialité NSI en terminale seulement, mais son utilisation, ne serait-ce que dans la modélisation de réseaux informatique est présente sur les trois années de lycée en SNT et NSI.

Les données structurées et leur traitement

Contenus	Capacités attendues
Données structurées	Identifier les différents descripteurs d'un objet. Distinguer la valeur d'une donnée de son descripteur. Utiliser un site de données ouvertes, pour sélectionner et récupérer des données.
Traitement de données structurées	Réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables.

La progression pour cette notion est relativement claire : **données structurées** à traiter sur tableur en seconde SNT, puis « Traitement de données en tables » avec un langage de programmation en première NSI et finalement « Bases de données » en Terminale NSI. En seconde, on pourra donc s'attacher à ce que les élèves sachent :

- * identifier les différents descripteurs d'un objet,
- * réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables.

Localisation, cartographie et mobilité

Une activité sur l'algorithme de Dijkstra peut permettre de travailler la notion d'algorithme indépendamment de la programmation.

Informatique embarquée et objets connectés

Partie importante, notamment en lien avec la programmation.
Mais sans apport d'entrée significative vis à vis de nos critères.

La photographie numérique

Partie importante, notamment en lien avec la programmation.
Mais sans apport d'entrée significative vis à vis de nos critères.

1. Les exercices

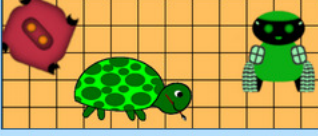
Une des difficultés pour aider nos élèves dans leur apprentissage, est la constitution d'une base d'exercices suffisamment nombreux et riches.

Concernant le nombre, on peut citer quelques exemples de support en ligne pouvant être proposés comme travail personnel aux élèves :

★ <https://parcours.algorea.org>

Plateforme développée par France-IOI. Elle propose un ensemble de contenus et activités interactives sur les thèmes du programme de seconde SNT (parcours SNT) et de la programmation (parcours Serious Game). Il n'y a pas besoin de créer de compte pour l'utiliser mais si l'élève veut garder la trace de sa progression il peut se créer un compte (ce sera le même compte que pour France-IOI).

ALGOREA SERIOUS GAME



- Apprendre à programmer en résolvant des puzzle game.
- Ce parcours est prévu pour couvrir le programme scolaire du collège. Notez que les premiers modules sont accessibles dès l'école élémentaire.
- Contenu pédagogique pour les enseignants disponible.
- Vidéo explicative, avec l'aimable autorisation de l'enseignant qui l'a conçue. Merci à lui !
- Adaptation du sujet de Brevet 2019 (exercice 4)
- Langage : **Scratch, Python**

SNT

Sciences du Numérique et Technologie

Ancienne version

Nouvelle version

- Découvrir SNT au lycée avec des activités interactives.
- Contenu pédagogique à destination des enseignants sur inscription au groupe Enseignants_SNT.

Si vous souhaitez accéder au contenu pédagogique qui accompagne le parcours SNT, vous pouvez nous demander le code d'accès en envoyant un mail à info@france-ioi.org

Parcours > Parcours officiels > SNT v2
 Plusieurs modules ont été développés grâce à l'aide et au soutien de plusieurs partenaires, comme Amazon Future Engineer et l'université de Lille.



Photographie numérique

Du pixel à la compression d'images et aux réglages d'un appareil photo numérique.



Localisation, cartographie et mobilité

Des cartes numériques au fonctionnement du GPS et des calculs d'itinéraires.



Les données structurées et leur traitement

Recherches dans des tables de données, manipulation de formats comme CSV ou JSON



Informatique embarquée et objets connectés

Programmer vos propres objets connectés !



Internet

(en cours de développement)



Le Web

(en cours de développement)



Les réseaux sociaux

(en cours de développement)

2 Travail sur la mise en œuvre

Parcours > Parcours officiels > ALGOREA SERIOUS GAME

- 1 - Découvrir comment donner des ordres à un robot**
Prise en main des différents types de défis.
Notion de séquence d'instructions
- 2 - Donner plusieurs fois le même ordre au robot**
Notion de boucle
- 3 - Donner plusieurs fois la même séquence d'ordres au robot**
Boucles à plusieurs instructions
Détection de motifs
- 4 - Concevoir des programmes compacts**
Notion d'imbrication
- 5 - Faire des choix en fonction des éléments de la grille**
Notion de condition
- 6 - Garder de l'information en mémoire**
Notion de variable
- 7 - Donner la même séquence d'ordres au robot jusqu'à ce qu'une condition soit remplie**
Notion de boucle conditionnelle
- 8 - Créer ses propres blocs**

Programmez le robot pour qu'il passe ramasser toutes les pierres précieuses puis rejoigne la case verte.

Vous pouvez avoir besoin de la boucle for

Fonctions disponibles : `tournerGauche()` fait se tourner le robot d'un quart de tour à gauche
Mots-clés autorisés : `for`

```
1 from robot import *  
2
```

PLUS DE DÉTAILS

Navigation: Play, Stop, Previous, Next, Volume

★ <http://www.france-ioi.org/algo/chapters.php>

Le site d'entraînement à la programmation et l'algorithmique. On peut y suivre la progression des élèves sous réserve de création d'un compte pour les élèves. Les niveaux 1 et 2 sont accessibles à tous, à partir du niveau 3, il faut débloquer les niveaux.

Cours et problèmes

Voir les cours et résoudre les problèmes en :

C	C++	Pascal	OCaml	Java	JavaScool	Python
---	-----	--------	-------	------	-----------	--------

Parcours général	Parcours lycée	Méthodes	Problèmes non classés
------------------	----------------	----------	-----------------------

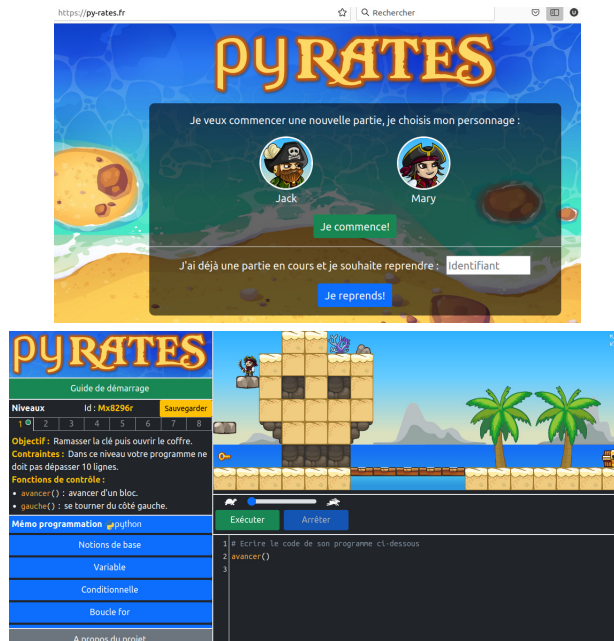
Le parcours général permet de progresser à son rythme, en partant des bases de la programmation et en allant jusqu'à un niveau avancé en algorithmique.

Niveau 1

› 1 - Affichage de texte, suite d'instructions	6 problèmes
› 2 - Répétitions d'instructions	13 problèmes
› 3 - Calculs et découverte des variables	10 problèmes
› 4 - Lecture de l'entrée	10 problèmes
› 5 - Tests et conditions	8 problèmes
› 6 - Structures avancées	8 problèmes
› 7 - Conditions avancées, opérateurs booléens	10 problèmes
› 8 - Répétitions conditionnées	5 problèmes

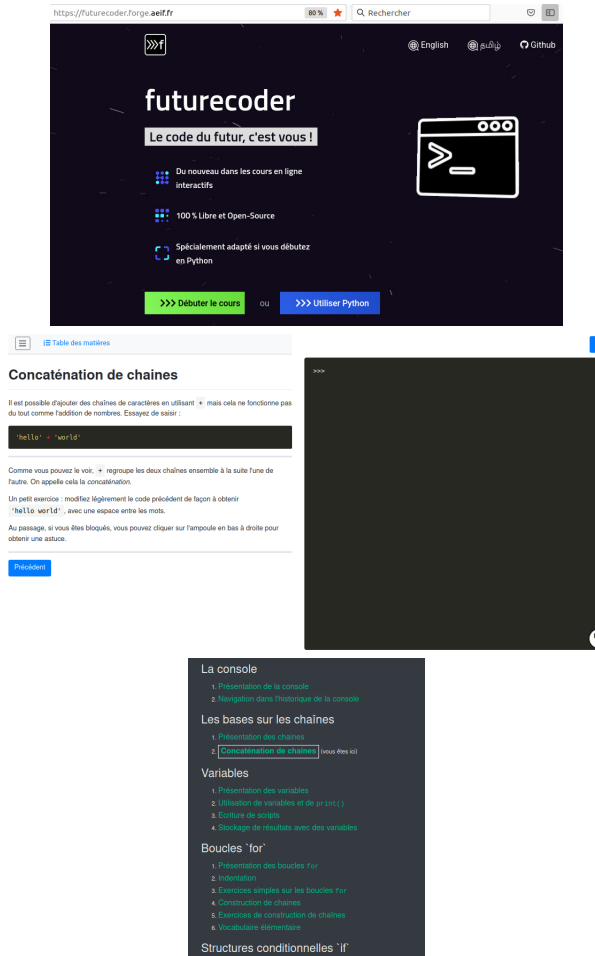
★ <https://py-rates.fr/>

Ce jeu met en scène les aventures d'un ou d'une pirate que vous avez choisi. Chaque niveau du jeu doit lui permettre d'ouvrir un coffre et d'en récupérer les pièces d'or. Le jeu contient 8 exercices.



★ <https://futurecoder.forge.aeif.fr/>

Un projet 100% open source d'envergure internationale de découverte de Python. C'est RGPD compatible. Il s'agit d'un cours complet pour débutant, avec des exercices interactifs, non mathématiques. Les élèves débutent directement avec les chaînes de caractères. Le parcours est plus difficile que la parcours algorithme.



Concernant la richesse, on peut citer :

★ **La trace de programme**

Cela permet aux élèves de lire des programmes.

Exemple de programme :

```

1 x = 3
2 y = 5
3 x = y - x
4 y = y - x
5 x = x + y
    
```

Trace attendue :

ligne	x	y
1	3	/
2	3	5
3	2	5
4	2	3
5	5	3

Vers une preuve :

ligne	x	y
	X	Y
3	$Y - X$	Y
4	$Y - X$	X
5	Y	X

Elle se fait par écrit de façon systématique. Si on fournit les tableaux, il ne faut pas les préremplir.

★ **les puzzles Parson**

Des lignes de code sont données avec éventuellement des lignes en trop et sans indentation. L'élève doit les réarranger pour satisfaire à la consigne.

Exemple :

<https://www.codepuzzle.io/P7A6W>



```

print('Vous êtes mineur')
print('Vous êtes majeur')
elif :
if age < 18 :
else :
elif age > 18 :
if age > 18 :
elif age < 18 :
    
```

Et la solution attendue :

```

if age < 18 :
    print('Vous êtes mineur')
else :
    print('Vous êtes majeur')
    
```

★ **L'apprentissage par étiquetage explicite (design pattern)**

Il s'agit d'expliciter en les nommant les sous-étapes des parties génériques d'un programme. Ces parties sont choisies pour exemplifier les stratégies classiques et récurrentes de résolution de sous-problèmes. De plus on distingue la lecture et de l'écriture de ces programmes.

Affectation

```
1 somme = terme + autre
```

Lecture

Écriture

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. Évaluer l'expression de droite du =. 2. Supprimer la valeur de la variable à gauche du =. 3. Assigner la valeur à la variable. | <ol style="list-style-type: none"> 1. Déterminer l'expression puis l'écrire en python. 2. Déterminer la variable et son nom en python. 3. Vérifier les types et écrire l'affectation. |
|--|---|

Instruction conditionnelle

```
1 if expression_bouleeenne : # Condition
2     action_si_vrai
3 else : # Tous les autres cas
4     action_si_faux
```

Lecture

Écriture

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. Identifier chaque bloc d'instructions. 2. Évaluer les expressions booléennes (les conditions) jusqu'à trouver True ou passer au bloc else. 3. Tracer le bloc correspondant. | <ol style="list-style-type: none"> 1. Distinguer les cas mutuellement exclusifs. 2. Ordonner et écrire les condition avec if, éventuellement des elif et un else. 3. Indenter chaque bloc conditionnel. |
|---|--|

Boucle non bornée

```
1 variable = valeur0 # Initialisaion
2 while condition(variable) : # Condition
3     variable = modif(variable) # Mise à jour
```

Lecture

Écriture

- | | |
|---|---|
| <ol style="list-style-type: none"> 1. Évaluer la condition. 2. Vrai : tracer le bloc et recommencer en 1. 3. Fausse : fini. | <ol style="list-style-type: none"> 1. Écrire : la condition de continuation ; 2. l'initialisation (avant) de la variable impliquée ; 3. et sa mise à jour (dans la boucle). |
|---|---|

★ **Écriture complète de programme**

C'est une tâche complexe. Il convient de différencier l'algorithme qui est l'idée du programme, du code qui est sa transcription dans un langage spécifique.

2. Le cours

La question est de savoir quelle trace de cours serait la plus à même d'aider nos élèves dans leur apprentissage. La question de la trace écrite en cours d'informatique n'est pas anodine, sachant qu'une part importante du travail va se faire devant un écran au détriment du papier, et que l'attente est autant dans le savoir-faire que les connaissances bruts.

Pistes de travail envisageables :

- ★ Travail complet sur notebook (avec éventuellement export pdf)
Cela fait beaucoup (trop ?) d'écran...
- ★ Utilisation réfléchie de vidéo
(Avec questionnaire et/ou demande de synthèse écrite... toujours beaucoup d'écran.)
- ★ Renvoi à un manuel ou cours papier plus classique.
Peut-être trop loin de la pratique et de la volonté des élève ?
- ★ Utilisation de flashcards.
C'est la piste suivie ci-après.

3. Propositions d'une mise en œuvre

Une idée

Notre idée était de partir sur une trace de cours minimale, un peu ludique, avec une utilisation la plus systématique possible et orientée, en SNT vers les connaissances discutées dans notre analyse.

Nous sommes partie sur l'idée des « *flashcard* » :

- * *Des fiches de questions (au recto) - réponses (au verso).*
- * *Chaque fiche ciblant*
 - *un savoir-faire (exercice corrigé avec rappel de cours)*
 - *ou un savoir (question de cours et cours)*
- * *Le tout assemblé dans un petit carnet complété au fil de l'année.*

En pratique

Construction

Il nous semble que la construction complète ou au moins à l'avance par thème devrait être faite préalablement à son introduction en classe. On pourrait donner les questions au fil des séances aux élèves (à coller ou recopier sur une page du carnet), charge à eux de construire les réponses (en ou hors classe) qui seraient ensuite et validée par l'enseignant avec une phrase « cours » alors ajoutée.

Utilisation

- * *(Orientation.) Temps d'institutionnalisation ciblant les savoirs et savoir-faire « Pour na NSI ».*
- * *(Ritualisation.) Rituel de début de cours : questions flash*
- * *(Automatismes.) Asseoir les bases essentielles de la science informatique en identifiant et faisant travailler ses automatismes.*
- * *(Évaluation.) Point d'appui d'au moins une évaluation sommative par période.*

En construction...

(Une présentation possible à l'échelle 1 en dernière page.)

1. Programmation**Affectation****Question**

La variable a « contient » une valeur entière.

Comment écrire un programme qui multiplie cette valeur par 2 puis lui ajoute 3 et la multiplie par la valeur que contenait a au départ ?

Réponse

```

1 # Sauvegarde de la valeur
2 b = a
3 # Calculs
4 a = a * 2
5 a = a + 3
6 a = a * b

```

À retenir

Utiliser une variable permet de mémoriser (sauvegarder) une valeur.

Question

Après lesquels des programmes suivants, la valeur de a vaut-elle 16 ?

```

1 a = 15
2 a = a - 1
3 a = a + 1

```

```

1 a = 14
2 a + 1
3 a + 1

```

```

1 a = 14
2 a = a + 1
3 a = a + 1

```

Réponse

Après le 1 elle vaut 15.
Après le 2 elle vaut 14.
Après le 3 elle vaut 16.

À retenir

Affecter une valeur à une variable « écrase » l'ancienne valeur.

Question

Après le programme suivant : quelle est la valeur de a ?

```

1 a = 15
2 b = 16
3 b = a
4 a = b

```

Réponse

La valeur de a et celle de b est 15.

À retenir

Affecter une valeur à une variable provoque une perte totale de son ancienne valeur.

Question

Quelle instruction permet d'ajouter 1 à la valeur de a ?

```
>>> a + 1
```

```
>>> a = a + 1
```

```
>>> a = + 1
```

```
>>> b = a + 1
```

Réponse

La valeur de a n'est modifiée que si la variable a précède un symbole =.
C'est la réponse : a = a + 1

À retenir

Le symbole = se lit « prend la valeur ».

Son action s'effectue de droite à gauche en évaluant la valeur à droite et en l'affectant à la variable écrite à gauche.

Question

Quelle est la signification de la ligne 2 ?

```
>>> a = 15
```

```
>>> a = a + 5
```

Réponse

La valeur de a est modifiée, a prend la valeur 20.

À retenir

Le symbole = se lit « prend la valeur ».

Son action s'effectue de droite à gauche en évaluant la valeur à droite et en l'affectant à la variable écrite à gauche.

Boucles

Question

Quelles sont les valeurs stockées dans les variables à la fin du programme ?

```

1 a = 1
2 b = 4
3 for i in range(5):
4     a = a + b
5     b = b + 3
    
```

Réponse

i	0	1	2	3	4
a	5	13	23	36	52
b	7	10	13	16	19

a vaut 52 et b vaut 19

À retenir

On évalue la valeur d'une variable avec les valeurs des variables actualisées

Question

Quelles sont les valeurs stockées dans les variables à la fin du programme ?

```

1 a = 10
2 b = 7
3 while a < 50 :
4     a = a + 10
5     b = a + b
    
```

Réponse

	V	V	V	V	F	
a	20	30	40	50	x	
b	27	57	97	147	x	

a vaut 50 et b vaut 147

À retenir

On évalue la valeur d'une variable avec les valeurs de variables actualisées.

L'exécution de la boucle `while` s'arrête dès que le test devient faux.

Instruction conditionnelle

Question

Dans le programme suivant, que se passe-t-il si à la ligne 4 on remplace `== par =` ?

```

1 a = 5
2 b = 7
3 if a == 10:
4     a = a + b
5 else :
6     b = a + b
    
```

Réponse

Le programme détecte une erreur, car après le `if`, il doit y avoir un test et `a = 10` n'est pas un test mais une affectation de valeur.

À retenir

Le symbole `=` est utiliser pour faire une affectation de valeur à une variable.

Le symbole `==` est utiliser pour faire un test dont le résultat est soit vrai (`True`) ou faux (`False`)

2. Internet

Question

Parmi les adresses suivantes lesquelles ne sont pas des adresse IP :

1. 192.1.4
2. 272.1.4.1
3. 192.168.1.2

Réponse

Seule la 3. est une adresse IP valide.

À retenir

Une **adresse IP**(v4), est constituée de quatre octets (séries de 8 bits) représentables par 4 entiers entre 0 et 255.

Question

À quoi sert le protocole TCP ?

Réponse

TCP (Transmission Control Protocol) définit la façon dont un message est assemblé en paquets avant d'être transmis sur internet et ré-assemblé dans le bon ordre à l'adresse de destination.

À retenir

Un **protocole** est un ensemble de règles, permettant l'échange d'informations.

3. Web

- * Différences entre Web et Internet
- * Hypertexte

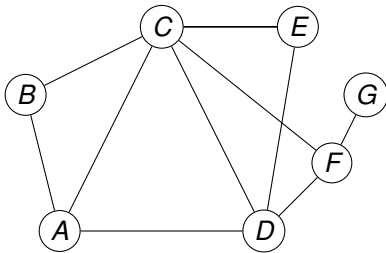
3 Exemples concrets

- * Expliquer les différentes informations dans les URL :
 - https://www.w3schools.com/python/python_conditions.asp
 - <http://www>.
- * Quel langage est utilisé ci-dessous ? (HTML/CSS)
- * A quoi sert le morceau de code suivant ?(HTML/CSS)
- * Quel est le rôle de chaque morceau de code (HTM+CSS)

4. Réseaux sociaux

Question

Pour le graphe suivant, déterminer son centre, son rayon et son diamètre.



Réponse

- * Centre : C ; D ou F
- * Rayon : 2
- * Diamètre : 3

À retenir

L'excentricité d'un sommet est la plus grande distance que le sépare de tout autre sommet.
Le rayon et le diamètre sont la plus petite et la plus grande excentricité.
Les centres sont les sommets d'excentricité égale au rayon.

5. Les données

- * Qu'est-ce qu'un fichier CSV ?
- * Questions de tableur...

6. Liste de questions à mettre en forme (+répondre + à retenir)

- * Quelle est la différence entre le WEB et Internet ?
- * Qu'est ce qu'une adresse IP ?
- * Qu'est ce qu'une url ?
- * En html, qu'est ce qu'une balise ?
- * Qu'est ce qu'un graphe ?
- * Qu'est ce qu'une donnée personnelle ?
- * Qu'est ce qu'un fichier csv ?
- * Si on a un tableau de données structurées, qu'est ce que filtrer les données ?
- * Si on a un tableau de données structurées, qu'est ce que trier les données ?

On considère le tableau de données suivant :

	B	C	D	E	F	G	H	I	J	K	L	M	N
1	SOCIETE	SECTEUR	Matériels_07	Logiciels_07	CA_07	Matériels_08	Logiciels_08	CA_08	ADRESSE	VILLE	CODEPT	CODE_DEF	REGION
2	A.B.S. GARONNE	ELECTRONIQUE	53886	715 914	769 800	1 086 030	120 670	1 206 700	Rue JEAN MERMOZ	PAU	64000		64AQUITAINE
3	A.C.E.	MECANIQUE	143312	132 288	275 600	514 400	514 400	1 028 800	43 Rue DE VINASSE	DOUARNENEZ	29100		29BRETAGNE
4	A.D.T.	MECANIQUE	8750	166 250	175 000	223 110	24 790	247 900	30 BD VAUGIRARD	MOULINS	03010		03AUVERGNE
5	A.U.R.T.	BTP	49728	660 672	710 400	486 810	54 090	540 900	3 Rue LORDAT	ROISSY-EN-FRANCE	95700		95ILE-DE-FRANCE
6	ABC	BTP	257856	121 344	379 200	145 200	338 800	484 000	17 SQUARE DUTILLEUL	MONTLHERY	91310		91ILE-DE-FRANCE
7	ACTION	BTP	457226	1 621 074	2 078 300	490 480	122 620	613 100	13 Avenue MAURICE FAURE	HVERES	83400		83PROVENCE-ALPES-COTE D'AZUR
8	AIRCUBE	ELECTRONIQUE	108459	25 441	133 900	270 080	1 080 320	1 350 400	1 Boulevard SOLIDARITE	POULLIGUEN	44510		44PAYS DE LA LOIRE
9	ALLEGRO	BTP	276115	287 385	563 500	1 039 150	1 039 150	2 078 300	24 Rue ESPRIT DES LOIS	CHATENAY-MALABRY	92290		92ILE-DE-FRANCE
10	ARC INFORMATIQUE	ELECTRONIQUE	264860	58 140	323 000	14 600	58 400	73 000	7 Rue MONTESPAN	MOISDON-LA-RIVIERE	44520		44PAYS DE LA LOIRE
11	ARCHET	BTP	117250	57 750	175 000	405 120	945 280	1 350 400	51 Rue KIESER	MASSY	91300		91ILE-DE-FRANCE
12	ARDE	BTP	152932	141 168	294 100	173 100	173 100	346 200	47 Rue KELLERMANN	COLOMBES	92700		92ILE-DE-FRANCE
13	ARDI	MECANIQUE	443700	295 800	739 500	245 240	367 860	613 100	244 Avenue DE L'INFANTE	FRONTIGNAN	34110		34LANGUEDOC-ROUSSILLON
14	ARMOR	MECANIQUE	139964	178 136	318 100	183 000	122 000	305 000	288 Rue GEORGES CLEMENCEAU	GUERET	23000		23LIMOUSIN
15	ARNO	BTP	213698	148 502	362 200	831 320	1 246 980	2 078 300	41 / 45 QUAI ROOSEVELT	CRETEIL	94000		94ILE-DE-FRANCE
16	ARTHUR	BTP	20102.03	134 529	154 631	548 730	60 970	609 700	196 Avenue DU GENERAL	SAINT-GERMAIN-EN-LAY	78110		78ILE-DE-FRANCE
17	AUBERT	ELECTRONIQUE	1212792	231 008	1 443 800	95 740	382 960	478 700	146 Rue PARADIS	AGEN	47000		47AQUITAINE
18	AUTOMOBILE NORDE	ELECTRONIQUE	15168	364 032	379 200	478 700	0	478 700	18 Rue VAUGIRARD	CALAIS	62100		62NORD-PAS-DE-CALAIS
19	BALI	ELECTRONIQUE	195472	833 328	1 028 800	167 440	41 860	209 300	8 Rue ST CLAUDE	CHAMBERY	73000		73RHONE-ALPES
20	BARDOT	ELECTRONIQUE	0	290 400	290 400	707 000	0	707 000	120 Avenue DU MARECHAL	ALENCON	61000		61BASSE-NORMANDIE
21	BELET	MECANIQUE	128920	105 480	234 400	70 000	105 000	175 000	28 Rue FERNAND CHRIST	CONDOM	32100		32MIDI-PYRENEES
22	BENAR	BTP	246840	43 560	290 400	102 880	925 920	1 028 800	15 Rue ARTHUR	DRANCY	93700		93ILE-DE-FRANCE
23	BENOIT	MECANIQUE	6582	103 118	109 700	430 830	47 870	478 700	37 BIS Avenue HONORE	MONTLUCON	03100		03AUVERGNE
24	BERGER	MECANIQUE	129756	66 844	196 600	349 140	814 660	1 163 800	14 Rue AUGUSTE COMTE	TOURS	37200		37CENTRE
25	BERNARD SA	MECANIQUE	31515	1 018 985	1 050 500	294 100	0	294 100	17 Avenue DE CUCILLE	SAINT-QUENTIN	02100		02PICARDIE
26	BIBLIO	BTP	211002	706 398	917 400	236 480	59 120	295 600	4 Rue BERLIER	AVIGNON	84000		84PROVENCE-ALPES-COTE D'AZUR

3 Exemples concrets

- * *On part du tableau, quel(s) filtre(s) peut on utiliser pour obtenir le tableau contenant uniquement les sociétés de la région Pays de la Loire ?*
- * *On part du tableau, quel(s) filtre(s) peut on utiliser pour obtenir le tableau contenant uniquement les sociétés spécialisée en électronique ?*
- * *On part du tableau, quel(s) filtre(s) peut on utiliser pour obtenir le tableau contenant uniquement les sociétés de la région Bourgogne spécialisée en mécanique ?*
- * *On part du tableau, quel(s) filtre(s) peut on utiliser pour obtenir le tableau contenant uniquement les sociétés de la région Bretagne ainsi que celle de la Aquitaine ?*
- *
- *
- *
- *
- * *Quelle est la différence entre un traitement de texte et un éditeur de texte ?*

(recto)

*La variable a « contient »
une valeur entière*

*Comment écrire un programme
qui multiplie cette valeur par 2
puis lui ajoute 3
et la multiplie par la valeur
que contenait a au départ?*

(verso)

```
1 # Sauvegarde de la valeur de a
2 b = a
3 # Calculs
4 a = a * 2
5 a = a + 3
6 a = a * b
```

*Utiliser une variable
permet de mémoriser
(sauvegarder) une valeur.*