Objectif numérique

Journée Programmation

PROFORMA

TECHNIFUTUR®
CENTRE DE COMPETENCES

Défi n°1

Histoire de programmation…

L'ordinateur est, contrairement aux apparences, très limité, il ne comprend que des instructions élémentaires. Le travail du programmeur est donc de décomposer chaque traitement qu'on souhaite lui voir accomplir en un flux d'instructions basiques, seules comprises de lui.

Consignes

Le Serious Game Algobot est un puzzle-game développé par le centre de compétences Technobel et la société Fishing Cactus avec des fonds publics de Digital Wallonia.

Téléchargement : https://www.algobot.be/

Le jeu consiste à programmer pas-à-pas un robot. Il doit se déplacer, parfois appuyer sur des boutons, prendre un colis ou le déposer.

Le panel d'actions possibles, à déplacer par glisser/déposer sur la ligne d'actions :

La ligne d'actions : il faut la compléter, instruction par instruction, pour que le robot puisse réaliser sa mission. La ligne F1 n'apparait pas en début de jeu mais sera bien nécessaire par la suite. F1 représente, en programmation, la notion de procédure, qui est un ensemble d'instructions que l'on peut exécuter plusieurs fois par la suite. Le nombre d'actions possibles de la ligne d'actions est limité, il vous faudra donc repérer les actions communes que l'on pourrait exécuter plusieurs fois et les placer dans F1. Ensuite mettre l'action F1 autant de fois que nécessaire dans la ligne d'actions. Une astuce, décomposer sur papier toutes les actions à entreprendre vous permettra de mieux cibler celles qui se répètent plusieurs fois.

Le panneau de commande : pas à pas en arrière, jouer la séquence d'actions, pas à pas en avant, le carré pour repasser en mode modification de la ligne d'actions

Défi

Votre premier défi vous permettra d'apprendre à décomposer un processus en n'utilisant que des instructions élémentaires. Il vous faudra aussi trouver le mot caché : le marquage est au sol, devant la sortie du niveau 1.9. Soyez attentif ! Ce mot vous permettra d'accéder aux défis suivant ☺

Bon amusement et que la force soit avec vous ☺

Défi n°2

Mais est-ce vraiment un défi ?

Histoire de programmation…

Nous allons ici vous parler de la méthode Agile. Il s'agit d'un nouveau mode de conduite de projets, lors du développement informatique d'une application. D'abord, le commanditaire du programme est régulièrement invité lors du processus, ainsi il peut apprécier l'évolution de son programme et éventuellement demander des changements en cours de route. Au moins, à la fin, le programme correspondra à ses attentes. L'application de cette méthode impacte aussi le mode de fonctionnement du groupe de programmeurs en charge du développement de l'application. Ils doivent collaborer, s'entraider en permanence, formaliser leurs difficultés, le besoin d'aide. En mode Agile, c'est le bon fonctionnement du groupe qui importe sur tout le reste et chacun doit y veiller. Les Soft-skills (capacité à communiquer, collaborer, …) sont donc des compétences essentielles dans le chef des programmeurs et recherchées par le monde de l'entreprise

Défi

Nous vous proposons donc, dans le cadre de cette journée d'initiation à la programmation, de vous exercer à l'application de la méthode Agile. L'entraide sera donc de mise, en faisant preuve de pédagogie, soutenir la réflexion de l'autre vaut mieux que de simplement donner la solution, l'appel à l'aide aussi.

Que l'esprit de collaboration soit avec vous ☺

Défi n°3

Histoire de programmation…

Avec Algobot, vous avez compris un peu mieux le travail du programmeur : décomposer des tâches complexes en instructions élémentaires. L'ordinateur les exécutera séquentiellement, c'est-à-dire les unes après les autres, du début jusqu'à la fin du programme. Cela nécessite l'apprentissage d'un raisonnement particulier de la part du programmeur : la logique de programmation, qui est fondamentale. Le programmeur a souvent besoin de coucher sur papier son raisonnement logique (autrement appelé algorithme) avant de l'encoder. Il le fait au moyen d'un langage informel, à l'aide d'un ordinogramme, ensemble de pictogrammes, qui permet de représenter toutes les instructions élémentaires que l'ordinateur peut comprendre, et que le programmeur doit maitriser.

Un programme a toujours 1 début et une fin. Les variables nous permettent de stocker en mémoire des données qui sont évolutives par simple modification de celles-ci. Elles portent chacune un nom donné par le programmeur et pour les manipuler, il suffit de mentionner celui-ci. Par exemple, si nous avons une variable km qui vaut 0 au départ, et si nous faisons une instruction km=km + 20, nous augmentons la valeur de km (au départ 0) de 20 et 20 sera la nouvelle valeur de km.

Nous avons également, comme instruction élémentaire, l'alternative, qui permet d'exécuter un bloc d'instructions plutôt qu'un autre grâce à l'évaluation d'une condition (expression logique qui renvoie vrai ou faux). Par exemple, si km> 100 alors je prends le train sinon je vais en vélo.

Défi

Dans ce défi, nous allons vous demander de formaliser ce petit problème de logique en ordinogramme.

Problème : Paul a reçu de l'argent de sa grand-mère. Il souhaite emmener sa copine en sortie, avec sa Vespa. Il commence par lui acheter des fleurs pour 13€ puis se rend chez elle, à 20 k de son domicile. Il souhaite l'emmener au resto à 5km, en prenant un menu chacun à 10 € et soit une bouteille de vin à 12€ ou alors une bière à 3€ l'unité, suivant ses moyens. Ensuite aller au cinéma, à 20 km de là, le prix de l'entrée est de 5€ la place. Ensuite, il reconduira sa copine en faisant un trajet de 20 km et si il a assez d'essence, il rentre chez lui, on alors il dormira chez sa copine.

Données du problème : Paul a 50€ et une Vespa. Son réservoir est rempli et, grâce à cela, il a autonomie de 90 km.

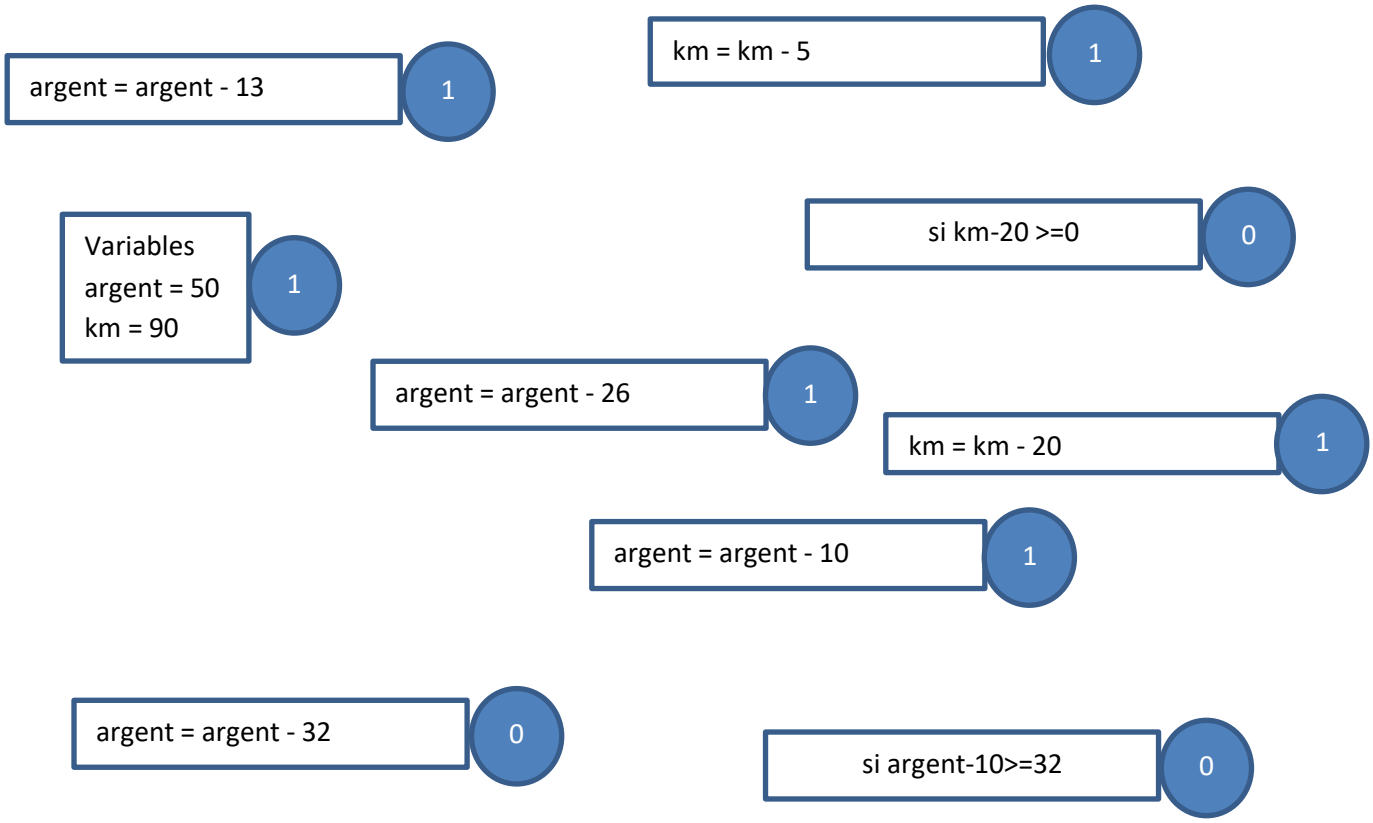
La question est la suivante : recomposer la logique en plaçant dans la stucture de l'ordinogramme qui vous est proposé les instructions ci-dessous. Chacune de celles-ci possède une valeur 0 ou 1. Une fois l'ordinogramme recomposé, réaliser la séquence de 0/1 de haut en bas et de gauche à droit si plusieurs instructions sont à la même hauteur. Vous composerez ainsi une valeur binaire composée de 12 bits (0 ou 1) qui vaut en décimal 3669. Google est votre ami ☺

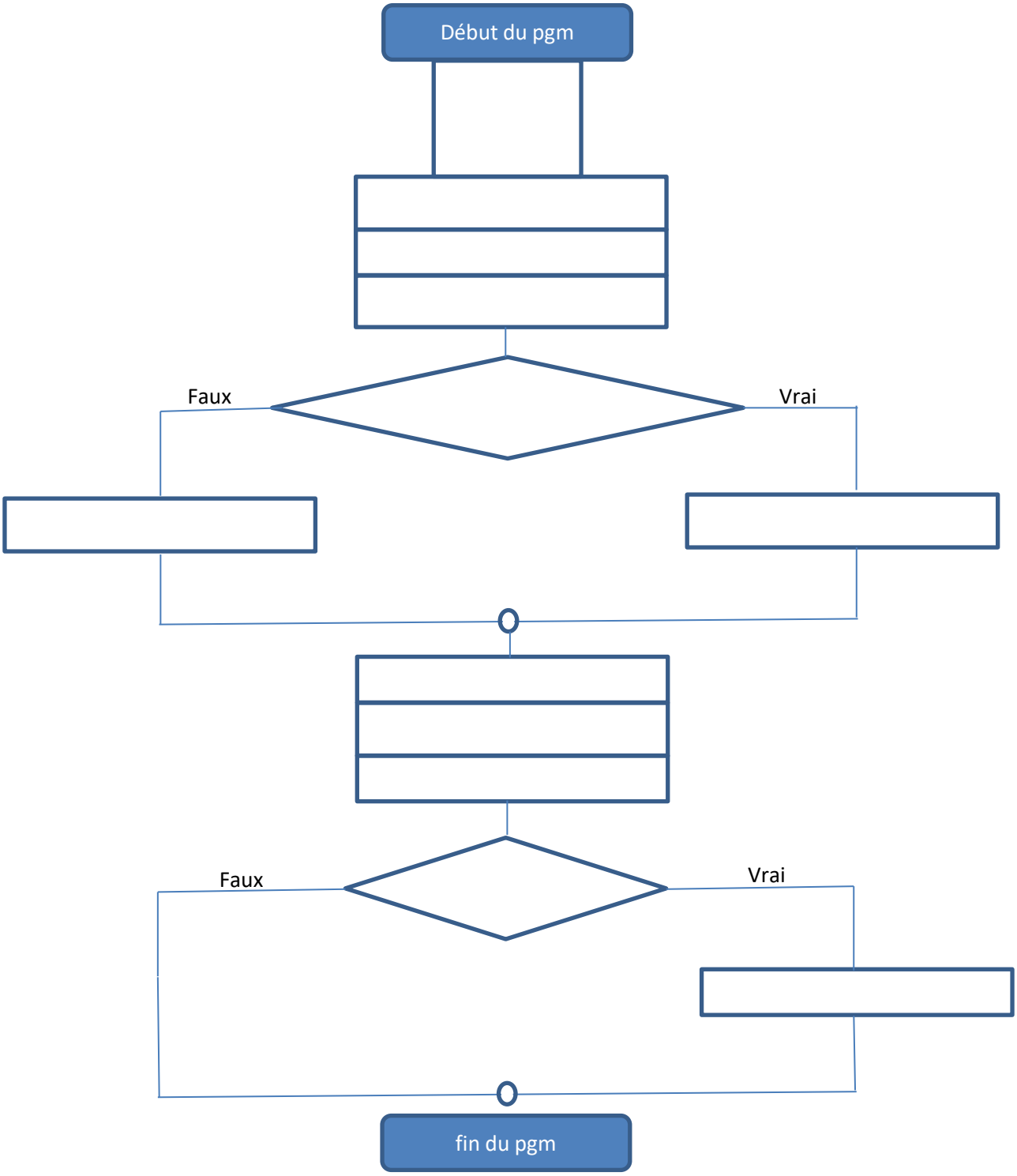
Histoire de programmation…

Pourquoi le binaire ? Nous comptons avec un système décimal (parce 10 symboles de 0 à 9, ensuite nous utilisons des règles pour compter plus haut en combinant les symboles : 10 à 99 puis 100 …).

L'ordinateur est composé de circuits électroniques basé sur des interrupteurs : le courant passe ou ne passe pas. Ces 2 états sont symbolisés par 1 ou 0 (appelés bit) d'où la notion de système binaire, système numérique de l'ordinateur. Avec 8 bits, on peut compter de 0 à 255 dans notre système décimal (00000000 à 11111111). Heureusement, alors qu'au début de l'informatique, le système binaire devait être compris des informaticiens cela est de moins en moins vrai. Pas peur donc ☺

Que la logique soit avec vous ☺




Début du pgm
Faux
Vrai
Faux
Vrai
fin du pgm

Défi n°4

Histoire de programmation…

Et si nous abordions la notion de langage de programmation ?

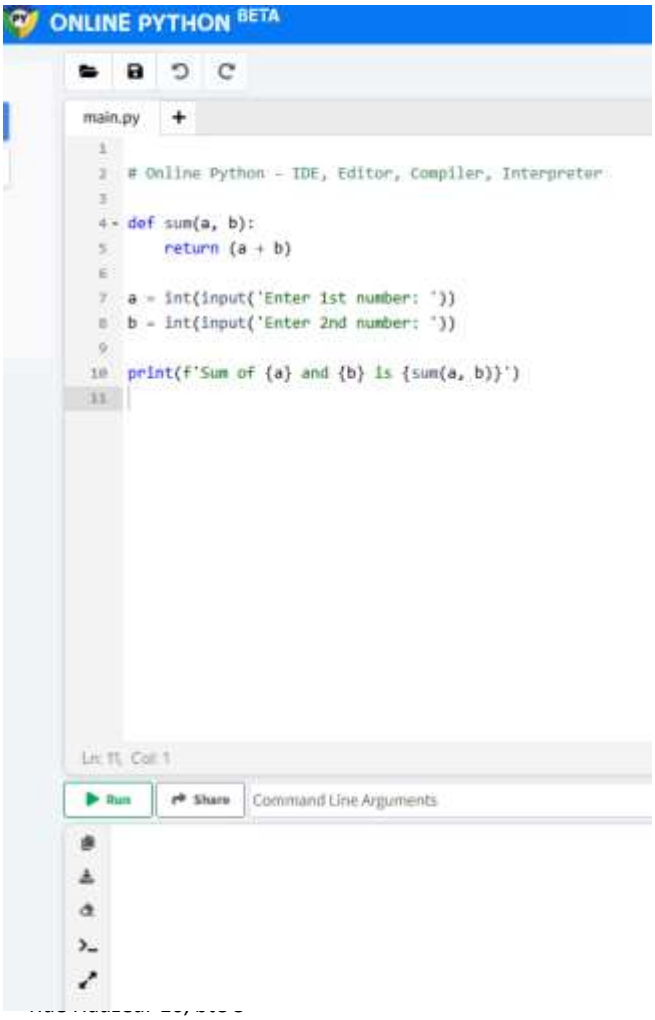
Un algorithme doit être encodé dans un langage de programmation. Pourquoi ? Parce que l'ordinateur ne comprend que le langage machine (binaire, que des 0 et des 1). Pas question pour les programmeurs de dialoguer directement en langage machine avec l'ordinateur. Nous possédons des langages de programmation, qui traduisent nos instructions en langage machine pour nous. Si la majorité des langages nous proposent des instructions en anglais, c'est un moindre mal ☺ Inutile de vous dire que la connaissance de l''anglais technique est bienvenue, tant pour la compréhension des mots des langages de programmation, que la formation continuée dans le domaine de la programmation, car la plupart des tutos/forums sont en anglais.

Nous vous proposons de formaliser l'algorithme précédent dans le langage de programmation Python.

Consignes

Pour cela, vous allez devoir aller sur le site : https://www.online-python.com/

L'interface se présente comme suit :



Le bouton + permet de créer un nouveau programme

Les lignes sont numérotées pour montrer l'ordre d'exécution des instructions

Exemple d'instruction (ligne 7 et 8) qui demande à l'utilisateur de rentrer une donnée qui sera placée dans la variable a, idem pour b. Attention, la donnée doit être un entier.

L'instruction print (ligne 10) affiche un message à l'écran. Si on veut mentionner l'affichage du contenu d'une variable, il fait mettre les { }. Les quotes ou apostrophes (sous la touche 4) sont obligatoires : il s'agit du délimiteur de chaines de caractères (ensemble de caractères qui ne sera pas analysé par l'ordinateur) Attention à notre langue française. Si dans notre message, il y a un quote, il faudra le faire précéder d'un \ pour dire que ce n'est pas la fin de la chaine.

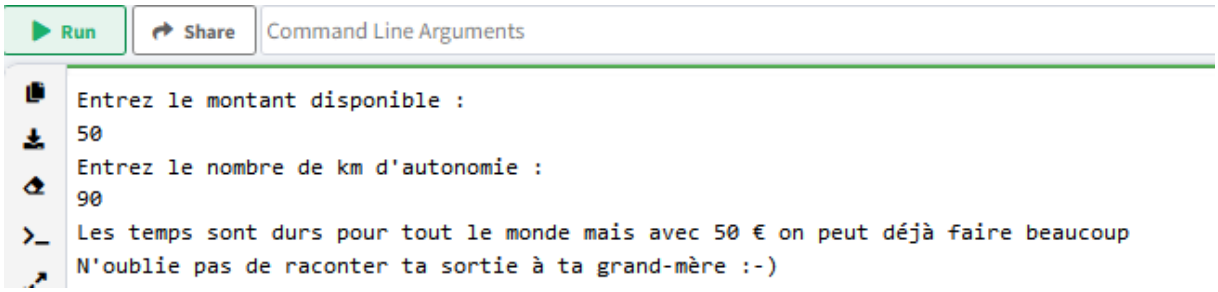
Pour exécuter le programme, cliquer sur Run

4031 Angleur
04/2249600

Créer un nouveau programme (onglet +) et encoder le début de programme suivant. Il vous montre aussi comment réaliser une alternative en Python. Attention l'indentation (décalage par rapport à la marge) est nécessaire pour indiquer à Python ce qui doit être exécuté si l'expression logique donne un vrai (après le if) ou donne un faux (après le else). Dans l'exemple, il n'y a qu'une seule instruction, mais il pourrait y en avoir plusieurs (et de toute nature).

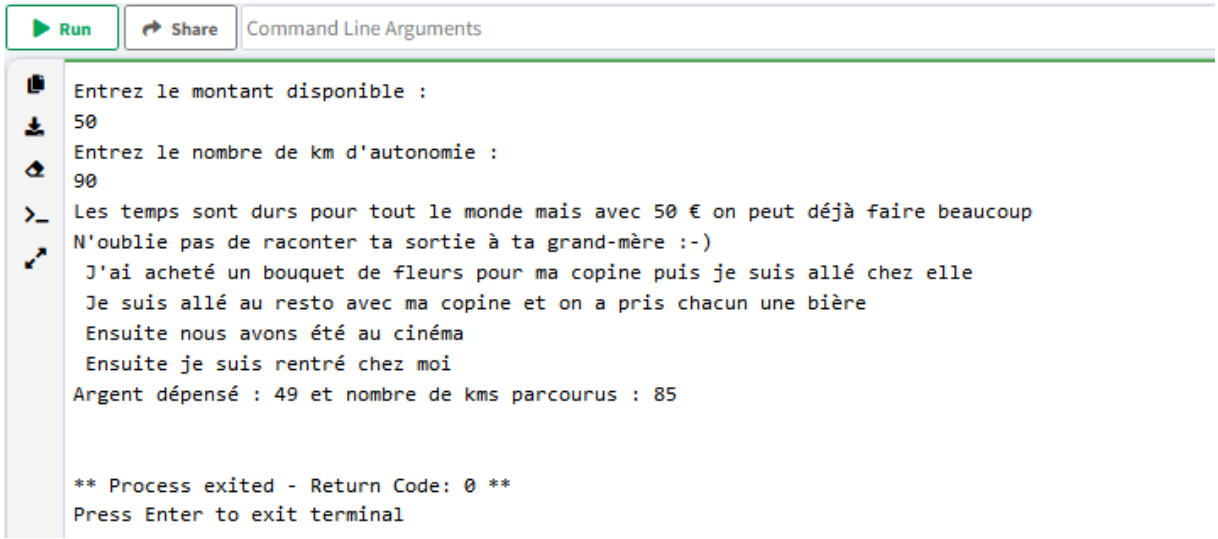
```
1  somme = int(input('Entrez le montant disponible :'))
2  autonomie = int(input('Entrez le nombre de km d\'autonomie :'))
3  if somme>=100:
4      print(f'Grand-mère est généreuse, elle t\'a donné {somme} €')
5  else:
6      print(f'Les temps sont durs pour tout le monde mais avec {somme} € on peut déjà faire beaucoup')
7  print(f'N\'oublie pas de raconter ta sortie à ta grand-mère :-)')
```

Voici ce que vous devez obtenir :

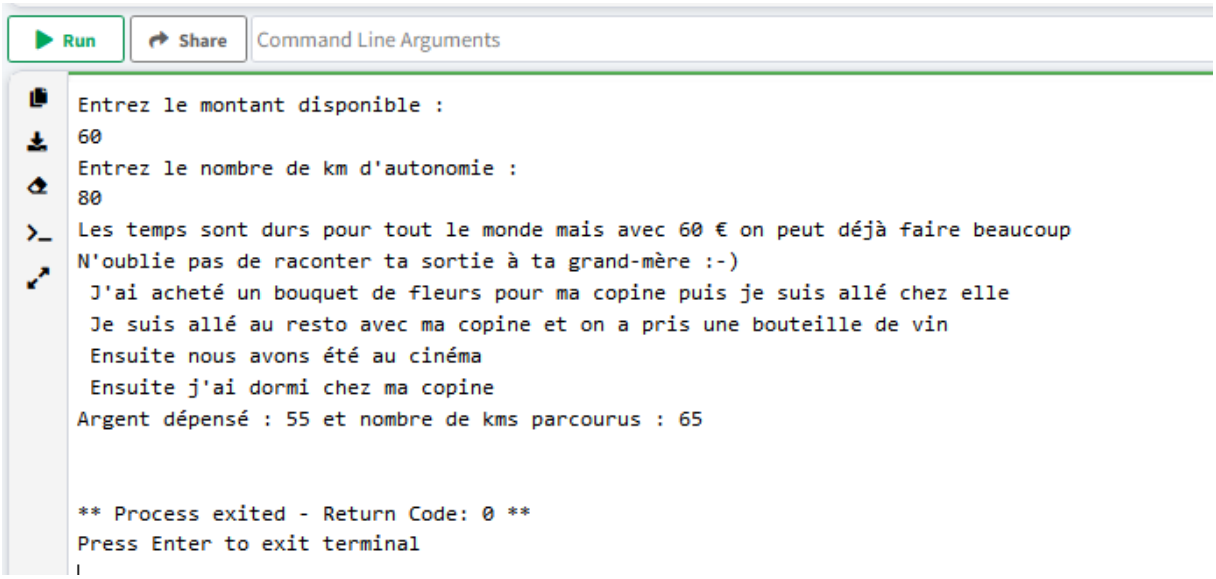


Défi

Réaliser le programme, sur base de la logique du défi 3, qui affiche le résultat suivant avec les données 50€ et 90 km. Attention, l'affichage final exprime bien la somme dépensée et le nombre de km parcourus (et pas le reste d'argent et le nombre de km restant possible).



Et ceci, si les données entrées sont 60€ et 80 km.



Que le code soit avec vous ☺

Défi n°5

Histoire de programmation…

Les bugs (erreur dans le programme) sont la hantise des programmeurs. Et quand cela arrive chez le client, cela ne fait vraiment pas sérieux. C'est pour cela qu'une phase de testing est toujours nécessaire. Les erreurs de syntaxe (instruction mal formulée dans le programme) sont à différencier des erreurs fonctionnelles (le traitement réalisé n'est pas celui attendu). Pour autant, le travail du programmeur consiste aussi à traiter les cas limites, par exemple, que fait votre programme si les données entrées sont 0 € et/ou 0 km ?

Et bien, Paul dépense de l'argent qu'il n'a pas et roule avec sa Vespa sans essence ☺.

Finalement, sa priorité serait les fleurs et les apporter à sa copine, puis aller au cinéma, si c'est possible, et si en plus il y a un resto, ce serait top.

Une astuce : on peut construire des expressions logiques (qui donnent un vrai ou un faux) complexes avec l'opérateur logique AND. Si les 2 membres de l'opérateur sont vrai, alors le résultat sera vrai, sinon ce sera faux. Exemple : si argent >= 13 et km >=20 alors je peux acheter des fleurs et les apporter à ma copine sinon je reste chez moi et je lui téléphone pour lui dire bonjour.

Défi

Modifier le programme en conséquence et tester avec différentes valeurs.

Avec 0€ et 0 km d'autonomie, vous devriez avoir :

```
Entrez le montant disponible :
0
Entrez le nombre de km d'autonomie :
0
Les temps sont durs pour tout le monde mais avec 0 € on peut déjà faire beaucoup
N'oublie pas de raconter ta sortie à ta grand-mère :-)
 Je vais me contenter de téléphoner à ma copine, pas de fleurs et pas assez d'essence pour une sortie
Argent dépensé : 0 et nombre de kms parcourus : 0
```

Avec 13€ et 20 km d'autonomie :

```
Entrez le montant disponible :
13
Entrez le nombre de km d'autonomie :
20
Les temps sont durs pour tout le monde mais avec 13 € on peut déjà faire beaucoup
N'oublie pas de raconter ta sortie à ta grand-mère :-)
 J'ai acheté un bouquet de fleurs pour ma copine puis je suis allé chez elle
Pas de resto pour cette fois-ci
 Pas de cinéma non plus
 Ensuite j'ai dormi chez ma copine
Argent dépensé : 13 et nombre de kms parcourus : 20
```

Avec 23€ et 100 km d'autonomie :

```
Entrez le montant disponible :
23
Entrez le nombre de km d'autonomie :
100
Les temps sont durs pour tout le monde mais avec 23 € on peut déjà faire beaucoup
N'oublie pas de raconter ta sortie à ta grand-mère :-)
 J'ai acheté un bouquet de fleurs pour ma copine puis je suis allé chez elle
Pas de resto pour cette fois-ci
 Ensuite nous avons été au cinéma
 Ensuite je suis rentré chez moi
Argent dépensé : 23 et nombre de kms parcourus : 80
```

Ce ne sera pas simple, que l'intelligence collective (du groupe) soit avec vous ☺

Défi n°6

Histoire de programmation…

Pour l'instant, nous avons travaillé en console, via une interface textuelle.

Heureusement, les applications proposent des interfaces graphiques, beaucoup plus agréables. De plus en plus souvent les applications sont online, donc accessibles via le Web. Lorsque vous aller sur un site internet, l'interface graphique que vous visualisez est en fait composée de code HTML, qui propose le contenu (texte, image, boutons, …) et de CSS, en charge de la mise en page (couleur de fond, des caractères, positionnement des éléments sur la page, …). Le code de chaque page peut être vu en faisant click droit et afficher le code source, ceci pour les curieux.

Attention, ni HTML, ni CSS ne sont des langages de programmation, mais simplement du code interprété par le browser qui en fait le rendu. Mais ils deviennent tellement omniprésents dans le développement software, que nous avons voulu vous proposer un défi à ce sujet.

Pour information, la partie visuelle d'un site est le Front-end, la partie côté serveur est le Back-end.

Comment allons-nous procéder pour apprendre la matière nécessaire à la réalisation du défi? Par l'e-learning bien-sûr ! C'est le quotidien du programmeur : l'autoformation est de mise dans un métier où les technologies n'ont de cesse d'évoluer. Harassant, pas vraiment, quand on développe une passion pour le code, c'est vraiment amusant de découvrir des nouveautés. Mais c'est aussi une obligation de se maintenir à jour pour une gestion de carrière réussie. Ne jamais se laisser enfermer dans des technologies obsolètes, là est le credo du programmeur.

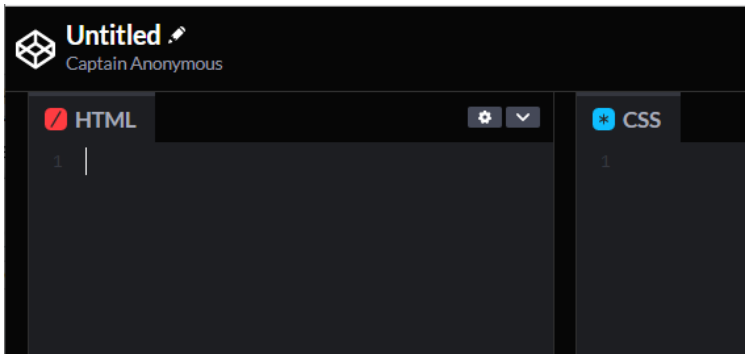
Consignes

Suivez ce tuto jusqu'au chapitre 5.

https://aymeric-auberton.fr/academie/html/

Dans ce tuto, il vous est demandé de créer des fichiers de code HTML (avec un logiciel de texte comme Notepad ++) et de les visualiser dans un navigateur de votre choix. Ce sera la technique à suivre en mode développement. En mode apprentissage, il y a plus simple. Il existe des sites qui vous permettent d'un côté de formaliser votre code et vous en avez tout de suite le rendu, tels qui serait affiché dans le navigateur. Simple, non ?

Nous vous proposons d'utiliser https://codepen.io/pen/



Il suffit de taper votre code dans la fenêtre de gauche, et vous verrez le rendu immédiatement en dessous. La fenêtre CSS, contiendrait, vous l'avez compris, le code CSS pour une mise en page.

Défi

Réaliser le code HTML qui donne le rendu suivant :

Une image a été intégrée ! Comment faire ? Google est votre ami. La propriété src de l'image est : https://flexboxfroggy.com/images/frog-green.svg

La couleur de fond (background du body) est PowderBlue. Comment faire ? Google est votre ami.

Un lien (balise a) renvoie sur le site de Wikipedia… Qui est votre ami ?

Que le génie créatif soit avec vous ☺