

Ergonomie : Rappels

Objectifs :

- Se préoccuper du point de vue de **l'utilisateur du programme**
- Améliorer la présentation générale du logiciel
- Faciliter l'utilisation du logiciel
- Gérer au mieux les Interactions Humain - Machine

Ergonomie : Définition

*Etude scientifique de la **relation entre l'humain et ses moyens, méthodes et milieux de travail** (sociologie, psychologie cognitive) et l'application de ces connaissances à la conception de systèmes "**qui puissent être utilisés avec le maximum de confort, de sécurité et d'efficacité par le plus grand nombre**"*

Recommandations ergonomiques :

- **Avoir une bonne organisation visuelle :**
 - Limiter la quantité d'informations à analyser
 - les messages et les figures de la console sont bien organisés
- **Être cohérent et respecter les conventions** au niveau
 - des appellations
 - de la présentation
- **Rendre l'information facilement disponible**
 - Fournir suffisamment d'informations
 - Prévoir des feedbacks
- **Être compréhensible**
- **S'adapter à l'utilisateur**
- **Gérer les erreurs des utilisateurs**
 - Empêcher si possible les erreurs
 - Permettre de repérer et comprendre les erreurs

Applications à un programme Python

Informations

les informations données à l'utilisateur doivent être compréhensibles et claires. Cela concerne :

- Le nom du programme
 - il doit indiquer ce que le programme fait
- le nom des dossiers et des sous dossiers :
 - images
 - son...
- **Présence d'un readme**
 - **Il indique:**
 - les auteurs
 - la licence
 - les bibliothèques à installer
 - les fonctionnalités du programme
 - Il est au format `.txt` ou `.md` (markdown)
 - surtout pas `.pdf` ou `.doc`

Guidage de l'utilisateur

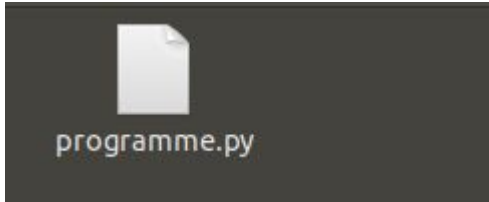
Lors du déroulement du programme, respecter **les critères ergonomiques**

- Information
 - menu
 - aide
- adaptation
 - choix de préférences
- gestions des erreurs
 - Le programme ne *plante* pas si l'utilisateur se trompe
- possibilités de quitter "*proprement*" le programme

Exemple : Calcul de périmètre

Version V1

Le dossier



Le programme

```
"""Calcul de périmètre"""
import math
def perimetre():
    """Le programme principal."""
    rayon = float(input())
    perimetre = 2 * math.pi * rayon
    print(perimetre)
```

L'exécution

```
4
25.132741228718345
```

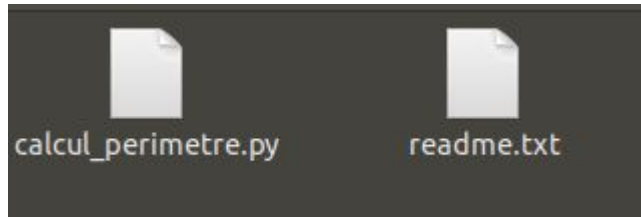
PROBLÈMES ERGONOMIQUE (informations)

- **Le nom du programme est mal choisi**
 - il n'indique pas ce que le programme fait
 - On le change :
`calcule_perimetre_cercle.py`
- **Pas de readme**
 - il faut en écrire un
- **Manque d'information et de guidage :**
 - message d'introduction
 - texte pour l'invite de saisie
 - message accompagnant le résultat
- **Mise en forme du résultat :**
 - trop de nombre après la virgule

Exemple : Calcul de périmètre

Version V2

Le dossier



Le readme

Programme de calcul de périmètre d'un cercle
 Démarrage : avec Python 3
 Auteur : Turing
 Licence : GPL

Le programme

```
"""Calcul de périmètre"""
import math
def perimetre():
    """Le programme principal."""
    print("Bienvenue : Ce programme
    calcule le périmètre d'un cercle à
    partir de son rayon")
    rayon = float(input("Entrez un rayon
    en cm "))
    perimetre = 2 * math.pi * rayon
    print("Le périmètre du cercle est de
    ",round(perimetre, 2), "cm")
```

L'exécution

Bienvenue : Ce programme calcule le périmètre d'un cercle à partir de son rayon
 Entrez un rayon en cm 4
 Le périmètre du cercle est de 25.13 cm

PROBLÈMES ERGONOMIQUE

Gestion des erreurs

Bienvenue : Ce programme calcule le périmètre d'un cercle à partir de son rayon
 Entrez un rayon en cm **quatre**

```
rayon = float(input("Entrez un
rayon en cm "))
ValueError: could not convert string to
float: 'quatre'
```

- Si l'utilisateur ne rentre pas un nombre, le programme s'arrête avec une erreur d'exécution !

Le programme

```
"""Calcul de périmètre"""
import math
def perimetre():
    """Le programme principal."""
    print("Bienvenue : Ce programme calcule le périmètre d'un cercle à partir de son rayon")
    rayon = None
    while rayon is None:
        saisie = (input("Entrez un rayon en cm "))
        if saisie.isnumeric() and float(saisie) > 0:
            rayon = float(saisie)
        else:
            print("Vous devez entrer un nombre positif.")
    perimetre = 2 * math.pi * rayon
    print("Le périmètre du cercle est de ", round(perimetre, 2), "cm")
```

L'exécution

```
Bienvenue : Ce programme calcule le périmètre d'un cercle à partir de son rayon
Entrez un rayon en cm quatre
Vous devez entrer un nombre positif.
Entrez un rayon en cm 4
Le périmètre du cercle est de 25.13
```

Exemple : Calcul de périmètre

Version V4 - V5 - V6...

Améliorations ergonomiques possibles

- ajout d'un menu
- ajout d'une aide disponible à tout moment
- Après la fin d'un calcul, donner la possibilité de recommencer un autre calcul ou d'arrêter le programme
- mettre de la couleur pour l'invite de saisie et pour l'affichage du résultat
- effacer l'écran lors d'un nouveau calcul
- dessiner une figure

Ajout de fonctionnalités

- Possibilité de choisir le nombre de chiffre après la virgule
- Possibilité de calculer le périmètre d'une autre figure (carré, rectangle ...)
- gestion des utilisateurs avec écriture et lecture dans un fichier pour se souvenir :
 - des réglages par défaut (le nombre de chiffres après la virgule, la couleur de l'affichage)
 - des calculs fait précédemment
 - etc...