Activité:

On souhaite construire de façon approchée la courbe représentative d'une fonction dérivable f qui vérifie $f'(a)=f(a)$ **pour tout réel a et** $f(0)=1$. Nous allons utiliser la méthode d'Euler qui repose sur l'utilisation d'une approximation affine d'une fonction en un point et donc construire point par point la courbe C_f .

Partie A : La méthode d'Euler

Étape 1 : $f(0)=1$ donc le point A(0 ; 1) se trouve sur C_f .

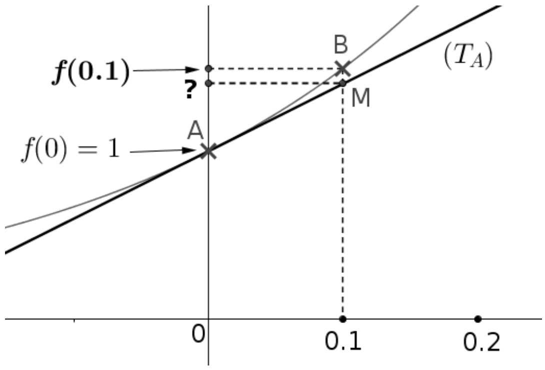
On cherche une approximation de $f(0{,}1)$.

Sur la figure ci-contre, B est le point de C_f d'abscisse 0,1.

Comme B est proche de A, la droite (AB) est proche de la tangente à C_f en A. Donc le coefficient directeur de la droite (AB) , $\dfrac{f(0{,}1)-f(0)}{0{,}1-0}$, est proche du coefficient directeur de (T_A) égal à $f'(0)$.

Donc $\dfrac{f(0{,}1)-f(0)}{0{,}1-0}\approx f'(0)$.

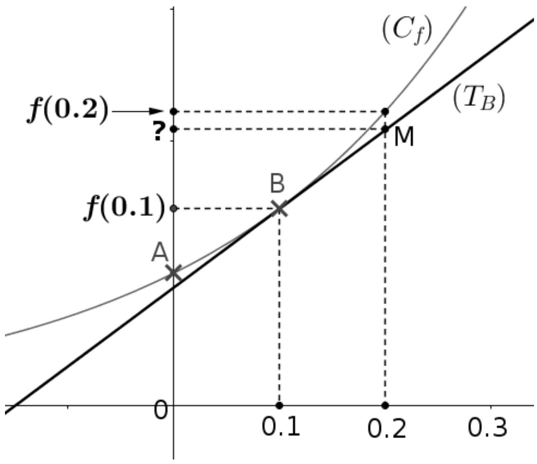
En utilisant cette approximation , donner une approximation de $f(0{,}1)$: ....



Étape 2 : Le point B(0,1 ; 1,1) appartient à C_f .

Comme précédemment, $\dfrac{f(0{,}2)-f(0{,}1)}{0{,}2-0{,}1}\approx f'(0{,}1)$.

En déduire une approximation de $f(0{,}2)$:



Étape 3 : Compléter le tableau. Si besoin, effectuer au brouillon les calculs pour les trois dernières colonnes ou mettre en évidence un moyen rapide de calculer les images demandées.

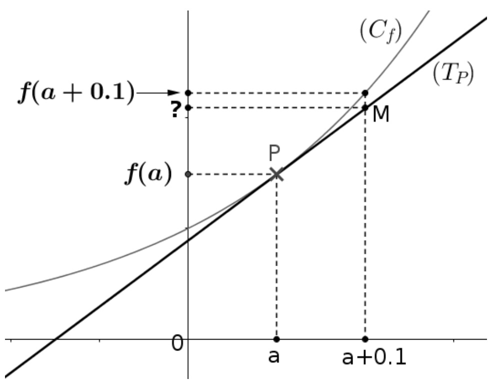
x	0	0,1	0,2	0,3	0,4	0,5
$y=f(x)$	1					

Étape 4 :

P(a ; $f(a)$) est un point précédemment trouvé sur la courbe C_f .

Quelle approximation va-t-on alors prendre pour $f(a+0{,}1)$?

Quelle approximation va-t-on alors prendre pour $f(a+h)$ en fonction de $f(a)$ et h ?



Partie B : La représentation graphique de la fonction recherchée

Ouvrir le fichier **Euler.py**

x et y sont deux listes qui désignent respectivement les abscisses et ordonnées des points appartenant à C_f .
Pour l'instant ces deux listes ne contiennent qu'un seul élément, les coordonnées de A.

```
x=[0]                #Liste qui va contenir les abscisses des points de Cf
y=[1]                #Liste qui va contenir les ordonnées approchées des points de Cf
```

a) En vous aidant de l'étape 4 de la partie A, compléter le programme ci-dessous et le saisir à partir de la ligne 4.

```
for i in range(40):              #On rajoute 40 points
    x.append(x[-1]+0.1)          #Abscisse du point suivant
    y.append(........................)       #Ordonnée approchée du point suivant
```

Exécuter le programme pour ainsi voir apparaître dans un repère ces 40 autres points placés à droite de A qui donnent l'allure de la courbe représentative de *f*.

b) En vous aidant de la fin de l'étape 4 de la partie A, rajouter les lignes de codes ci-dessous (à la ligne 8) qui permettent de compléter les listes x et y avec des valeurs négatives pour les abscisses.

```
x.append(0)                          #On repart du point A de coordonnées (0;1)
y.append(1)
for i in range(50):                  #On rajoute 50 autres points à gauche de A
    x.append(x[-1]-0.1)              #Abscisse négative du point suivant
    y.append(........................)       #Ordonnée approchée du point suivant
```

Exécuter le programme pour ainsi voir apparaître tous les points de coordonnées (x[i] ;y[i]).

c) Remplacer 0.1 par 0.01 dans tout le programme et augmenter le nombre d'itérations à effectuer dans chaque boucle **for i in range()** pour obtenir beaucoup plus de points.

d) Quel est l'élément de la liste y qui donne une valeur approchée de $f(1)$? ....

Afficher cette valeur : $f(1) \approx$................ (3 chiffres significatifs)