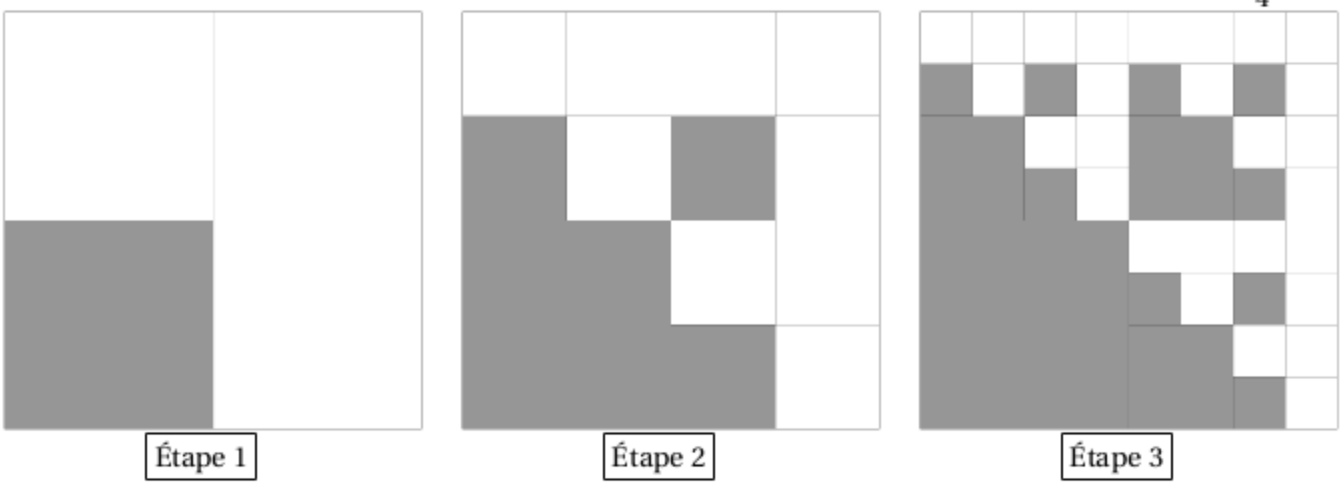
Capacités du chapitre 2

Capacité 1 ***Conjecturer la limite d'une suite définie par un motif géométrique***

On colorie un carré en plusieurs étapes :

- Étape 1 : on partage le carré en quatre carrés de même aire et on colorie le carré en bas à gauche;
- Étape 2 : on partage chaque carré non coloriée en quatre en quatre carrés de même aire et on colorie le carré en bas à gauche;
- Étapes suivantes : on répète le procédé avec chaque carré non colorié obtenu à l'étape précédente.

Pour tout entier $n \geqslant 0$, soit b_n la fraction du carré initial qui n'est pas coloriée à l'étape n, ainsi $b_1 = \frac{3}{4}$.



1. Pour tout entier $n \geqslant 0$, exprimer b_{n+1} en fonction de b_n et en déduire la nature de la suite (b_n).
2. Pour tout entier $n \geqslant 0$, déterminer une formule explicite de b_n.
3. Conjecturer avec la calculatrice si la suite (b_n) possède une limite finie.
4. Écrire une fonction Python qui retourne le nombre d'étapes nécessaires pour que 99% du carré initial soit colorié.

1) Pour tout entier $n \geqslant 0$, on a :

$$b_{n+1} = \frac{3}{4} \times b_n$$

On en déduit que la suite (b_n) est géométrique de raison $\frac{3}{4}$.

2) D'après une propriété du cours, pour tout entier $n \geqslant 0$: $b_n = b_0 \times \left(\frac{3}{4}\right)^n$

donc $b_n = \left(\frac{3}{4}\right)^n$ car $b_0 = 1$

3) Avec la calculatrice on peut conjecturer que $\lim\limits_{n \to +\infty} b_n = 0$

4)

```python
# Type your text here
#u(0) = 1
#u(n+1)= 0.75 * u(n)
# recherche du plus petit entier
#n tel que u(n) <= 0.01

def seuil(s):
  u = 1
  n = 0
  while u > s:
    u = 0.75 * u
    n = n + 1
  return n

print(seuil(0.01))
```

deg PYTHON

```
>>> from chapitre2_capacite1 -
17
>>> |
```

Relancer Sauvegarder

Capacité 2 *Utiliser la définition d'une suite convergente*

1. Soit la suite (u_n) définie pour tout entier $n \geqslant 1$ par $u_n = 1 - \frac{1}{n}$.

   a. La suite (u_n) est-elle monotone ? Justifier.

   b. Quelle limite peut-on conjecturer pour la suite (u_n) ?

   c. Déterminer à partir de quel rang, on a $u_n \in]-10^{-3}; 10^{-3}[$. $]1-10^{-3}; 1+10^{-3}[$

   d. À partir de la définition, démontrer que (u_n) converge.

2. Soit la suite (v_n) définie pour tout entier $n \geqslant 1$ par $v_n = 1 - \frac{(-1)^n}{n}$.

   a. La suite (v_n) est-elle monotone ? Justifier.

   b. Conjecturer la limite de (v_n) puis démontrer qu'elle converge à partir de la définition.

3. Soit la suite (w_n) définie pour tout entier naturel n par $w_n = (-1)^n$.

1) a) Démontrons que la suite $(u_n)_{n\geqslant 1}$ est croissante :

Pour tout entier $n \geqslant 1$:

$$u_{n+1} - u_n = 1 - \frac{1}{n+1} - \left(1 - \frac{1}{n}\right) = \frac{1}{n} - \frac{1}{n+1}$$

donc $u_{n+1} - u_n = \frac{1}{n(n+1)}$

donc $u_{n+1} - u_n > 0$

La suite $(u_n)_{n\geqslant 1}$ est donc strictement croissante

deg SUITES

Suites | Graphique | Tableau

Régler l'intervalle

n	u_n
1	0
2	0.5
3	0.6666667
4	0.75
5	0.8
6	0.8333333
7	0.8571429
8	0.875

b) Avec le tableur de la calculatrice, on peut conjecturer que (u_n) converge vers 1.

c) Déterminons s'il existe un plus petit entier naturel $n \geq 1$ tel que :

$$u_n \in]1-10^{-3} ; 1+10^{-3}[$$

On raisonne par équivalences :

$$u_n \in]1-10^{-3};1+10^{-3}[\Leftrightarrow 1-10^{-3} < 1-\frac{1}{n} < 1+10^{-3}$$

$$\Leftrightarrow 10^{-3}-1 > \frac{1}{n}-1 > -1-10^{-3}$$

$$u_n \in]1-10^{-3};1+10^{-3}[\Leftrightarrow 10^{-3} > \frac{1}{n} > -10^{-3}$$

On sait que n entier naturel non nul, donc :

$$u_n \in]1-10^{-3};1+10^{-3}[\Leftrightarrow 10^{-3} > \frac{1}{n} > 0$$

$$\boxed{u_n \in]1-10^{-3};1+10^{-3}[\Leftrightarrow 10^{3} < n}$$

A partir du rang $10^3+1 = 1001$, on aura $u_n \in]1-10^{-3};1+10^{-3}[$.

d) On démontre de même qu'en c) que pour tout réel $a>0$, l'entier n_a immédiatement supérieur à $\frac{1}{a}$ est telle que pour tout entier $n \geq n_a$, $u_n \in]1-a;1+a[$

Ainsi la suite (u_n) vérifie la définition d'une suite qui converge vers 1.

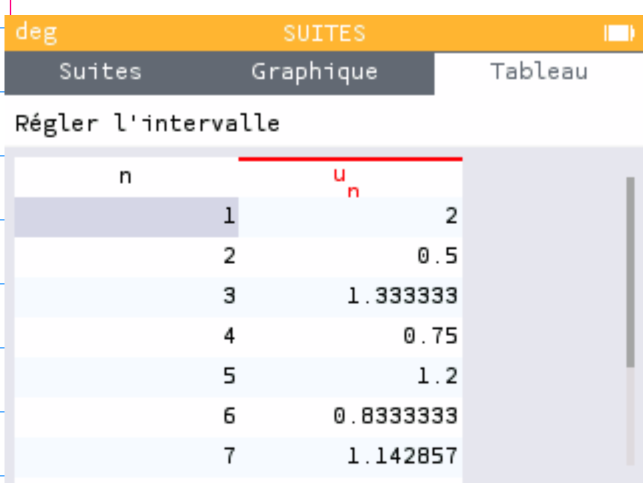
2) Soit (v_n) la suite définie pour tout entier $n \geq 1$ par : $v_n = 1 - \frac{(-1)^n}{n}$

a) Pour tout entier $n \geqslant 1$, on a :

$$v_{n+1} - v_n = \frac{(-1)^{n+1}}{n+1} - \frac{(-1)^n}{n} = (-1)^{n+1} \times \left(\frac{1}{n+1} + \frac{1}{n}\right)$$

donc $v_{n+1} - v_n = (-1)^{n+1} \times \frac{2n+1}{n(n+1)}$

On a $\frac{2n+1}{n(n+1)} > 0$ donc $v_{n+1} - v_n$ du signe de $(-1)^{n+1}$. Mais $(-1)^{n+1}$ n'est pas de signe constant, car son signe dépend de la parité de n. Ainsi $v_{n+1} - v_n$ n'est pas de signe constant et la suite $(v_n)_{n \geqslant 1}$ n'est pas monotone.



n	u_n
1	2
2	0.5
3	1.333333
4	0.75
5	1.2
6	0.8333333
7	1.142857
8	0.875

2) b) Avec le tableur de la calculatrice, on peut conjecturer que $(v_n)_{n \geqslant 1}$ converge vers 1

c) Démontrons que (v_n) converge vers 1

Soit a un réel strictement positif.

On raisonne par équivalences :

$v_n \in]1-a ; 1+a[\iff 1-a < 1 - \frac{(-1)^n}{n} < 1+a$

$v_n \in]1-a ; 1+a[\iff -a < \frac{(-1)^n}{n} < a$

On passe à la valeur absolue :

$v_n \in]1-a ; 1+a[\iff \left| \frac{(-1)^n}{n} \right| < a$

$\iff \frac{1}{n} < a$

$v_n \in]1-a ; 1+a[\iff n > \frac{1}{a}$

Ainsi pour tout réel $a > 0$, il existe un entier n_a immédiatement supérieur à $\frac{1}{a}$, tel que pour tout entier $n \geqslant n_a$:

$$v_n \in]1-a ; 1+a[$$

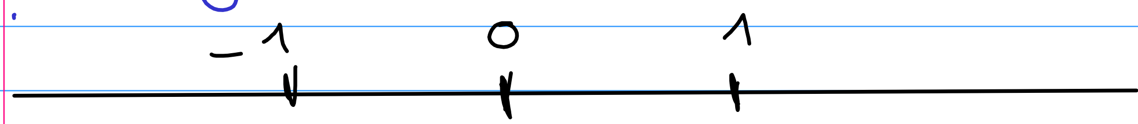
Ainsi la suite (v_n) vérifie la définition d'une suite convergeant vers 1.

3) Soit $(w_n)_{n \geqslant 0}$ la suite définie pour tout entier $n \geq 0$ par :

$$w_n = (-1)^n$$

Démontrons par l'absurde que la suite (w_n) ne converge pas vers un réel ℓ.

Hypothèse : on suppose que la suite (w_n) converge vers un réel ℓ.



Par définition il existe un entier $n_{0,5}$, tel que pour tout entier $n \geq n_{0,5}$, on a :

$w_n \in]\ell - 0,5 ; \ell + 0,5[$

En particulier on doit avoir :
$w_{n_{0,5}}$ et $w_{n_{0,5}+1}$ dans l'intervalle
$]\ell - 0,5 ; \ell + 0,5[$.

Or $|w_{n_{0,5}} - w_{n_{0,5}+1}| = 2$ car deux termes consécutifs de la suite ont pour valeurs -1 et 1.

On aboutit à une contradiction : la distance entre $w_{n_{0,5}}$ et $w_{n_{0,5}+1}$ est de 2 et ils doivent appartenir à $]\ell - 0,5 ; \ell + 0,5[$ d'amplitude 1.

Par conséquent l'hypothèse de départ est fausse et on peut affirmer que la suite (w_n) ne converge pas vers 1.

Capacité 4 ***Modélisation par une suite et algorithme de seuil***

Chaque semaine, un agriculteur propose en vente directe à chacun de ses clients un panier de produits frais qui contient une seule bouteille de jus de fruits. Dans un esprit de développement durable, il fait le choix de bouteilles en verre incassable et demande à ce que chaque semaine, le client rapporte sa bouteille vide.

On suppose que le nombre de clients de l'agriculteur reste constant.

Une étude statistique réalisée donne les résultats suivants :

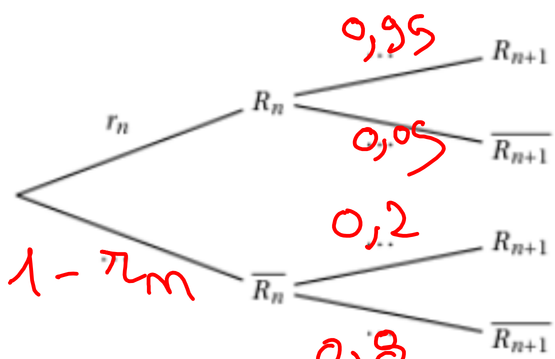
- à la fin de la première semaine, la probabilité qu'un client rapporte la bouteille de son panier est 0,6;

- si le client a rapporté la bouteille de son panier une semaine, alors la probabilité qu'il ramène la bouteille du panier la semaine suivante est 0,95 ;
- si le client n'a pas rapporté la bouteille de son panier une semaine, alors la probabilité qu'il ramène la bouteille du panier la semaine suivante est 0,2.

On choisit au hasard un client parmi la clientèle de l'agriculteur. Pour tout entier naturel n non nul, on note R_n l'évènement « le client rapporte la bouteille de son panier de la n-ième semaine ».

Pour tout entier naturel n non nul, on note r_n la probabilité que le client rapporte la bouteille du panier de la n-ième semaine. On a alors $r_n = \mathbb{P}(R_n)$.

1. Recopier et compléter l'arbre pondéré (aucune justification n'est attendue) :



2) Pour tout entier $n \geq 0$;

R_n et $\overline{R_n}$ forment une partition de l'univers donc d'après la formule des probabilités totales, on a :

$$P(R_{n+1}) = P(R_{n+1} \cap \overline{R_n}) + P(R_{n+1} \cap R_n)$$

On applique deux fois la formule des probabilités composées.

$$P(R_{n+1}) = P(\overline{R_n}) \times P_{\overline{R_n}}(R_{n+1}) + P(R_n) \times P_{R_n}(R_{n+1})$$

$$P(R_{n+1}) = (1-r_n) \times 0{,}2 + r_n \times 0{,}95$$

donc $P(R_{n+1}) = 0{,}2 + 0{,}75 r_n$

donc $\boxed{r_{n+1} = 0{,}2 + 0{,}75 r_n}$

3) Pour tout entier naturel $n \geqslant 1$, on définit la propriété : $\mathcal{P}_n$: " $r_n < 0{,}8$ "

Démontrons par récurrence que $\mathcal{P}_n$ est vrai pour tout entier $n \geqslant 1$:

Initialisation On a $r_1 = P(R_1) = 0{,}6$

donc $r_1 < 0{,}8$ donc $\mathcal{P}_1$ est vraie

Hérédité

Hypothèse de récurrence : Soit un entier $n \geqslant 1$ tel que $\mathcal{P}_n$ est vraie.

L'hypothèse de récurrence se traduit par :

$$r_n < 0{,}8$$

On applique la relation de récurrence :

$$0{,}75 r_n + 0{,}2 < 0{,}75 \times 0{,}8 + 0{,}2$$

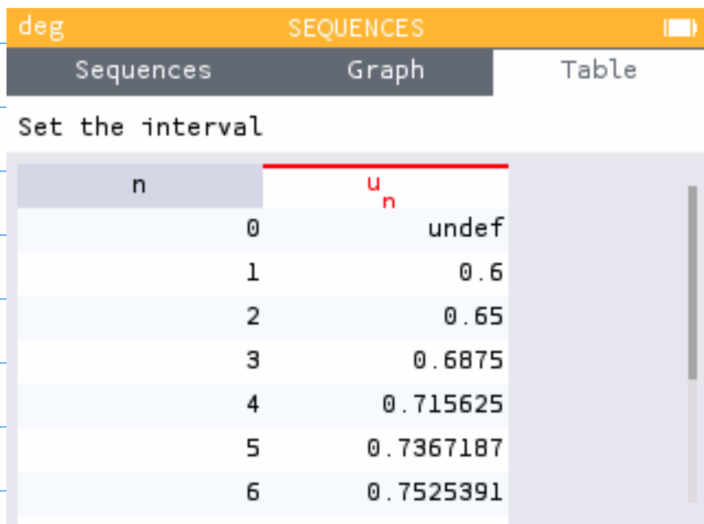
donc $r_{n+1} < 0,8$

donc P_{n+1} est vraie

donc la propriété est héréditaire.

Conclusion P_n est initialisée pour $n=1$ et elle est héréditaire donc elle est vraie par récurrence pour tout entier $n \geq 1$.

3)



n	u_n
0	undef
1	0.6
2	0.65
3	0.6875
4	0.715625
5	0.7367187
6	0.7525391
7	0.7644043

On peut conjecturer que la suite (r_n) converge vers 0,8.

5) On admet que (r_n) converge vers une limite ℓ.

Pour tout entier $n \geq 1$, on a :

$$r_{n+1} = 0,75\, r_n + 0,2$$

D'après une propriété du cours,

si $\lim_{n \to +\infty} r_n = \ell$ alors $\lim_{n \to +\infty} r_{n+1} = \ell$

Par règles opératoires de produit et de somme on a : donc en passant à la limite dans $r_{n+1} = 0{,}75 r_n + 0{,}2$:

$$\ell = 0{,}75 \times \ell + 0{,}2$$

$$\iff 0{,}25 \ell = 0{,}2$$

$$\iff \boxed{\ell = 0{,}8}$$

La suite (r_n) converge donc vers 0,8.

6. Justifier qu'il existe un entier n tel que $u_n > 0,79$ et compléter la fonction Python ci-dessous pour que `seuil(0.79)` retourne le plus petit entier n tel que $r_n > 0,79$. Il s'agit d'un **algorithme de seuil**.

```
def seuil(s):
    r = 0.9
    n = 1
    while r <= s:
        r = 0.75*r+0.2
        n = n + 1
    return n
```

Si $\lim_{n \to +\infty} r_n = 0{,}8$ alors par définition il existe un entier N tel que pour tout entier $n \geqslant N$, on a : $u_n \in]0{,}79 ; 0{,}81[$

Pour tout entier $n \geqslant N$, on a donc :

$$0,79 < u_n.$$

7. Modifier la fonction `seuil(s)` en une fonction `seuil2(s)` qui retourne le plus grand entier n tel que $u_n < s$. Pour quelles valeurs de s, l'exécution de `seuil2(s)` ne se terminera-t-elle pas ?

```
def seuil2(s):
    r = 0.9
    n = 1
    while r < s:
        r = 0.75*r + 0.2
        n = n + 1
    return n - 1
```

L'exécution de seuil2(s) ne se terminera pour toute valeur $s \geqslant 0.8$.