

```
pika@pika-ThinkPad-T480 ~/test tar -jcvf files  
tar: Je refuse de créer un fichier d'archive vide (oui je suis lâche)  
Pour en savoir davantage, faites : « tar --help » ou « tar --usage ».  
pika@pika-ThinkPad-T480:~/test php
```

RUSH 2 - KICKOFF

PHP, bash, systèmes de compression

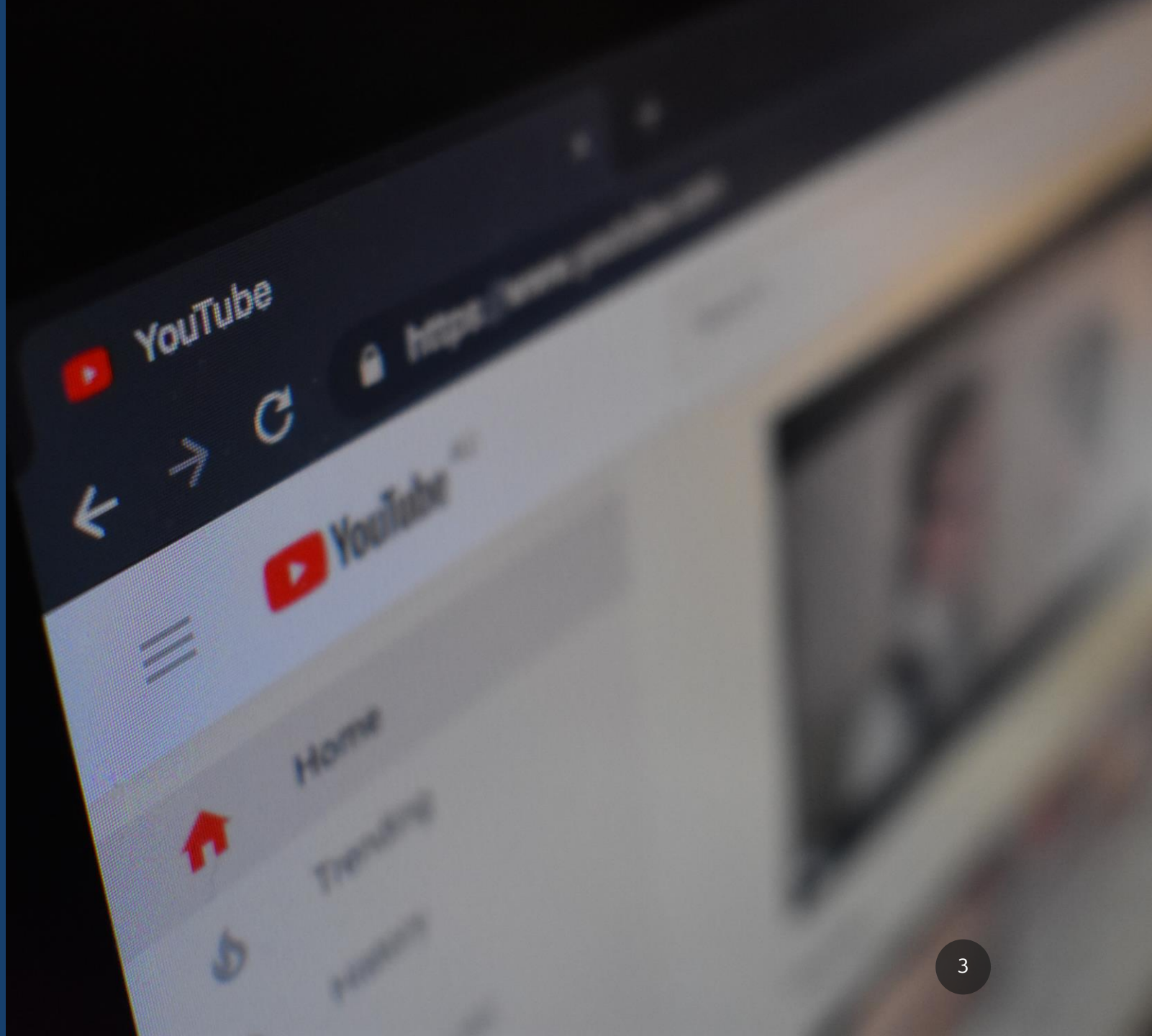
COMPRESSION DE DONNÉES

Si vous suivez les aventures de Pied Piper, vous savez que la compression est un domaine vital de la tech aux applications omniprésentes.



POURQUOI COMPRESSER

- Cloud (quantité importante d'information)
- Streaming (adaptation au trafic sur le réseaux pour une utilisation en temps réel)
- etc...



« COMPACTAGE » OU « COMPRESSION »

- Méthode réversible
- Conserve toute l'information
- Ex : Zip, Flac, etc...
- Méthode irréversible
- Perd une partie de l'information
- Ex : JPEG, mp3, etc...

Pourquoi accepter de perdre de l'information... ?

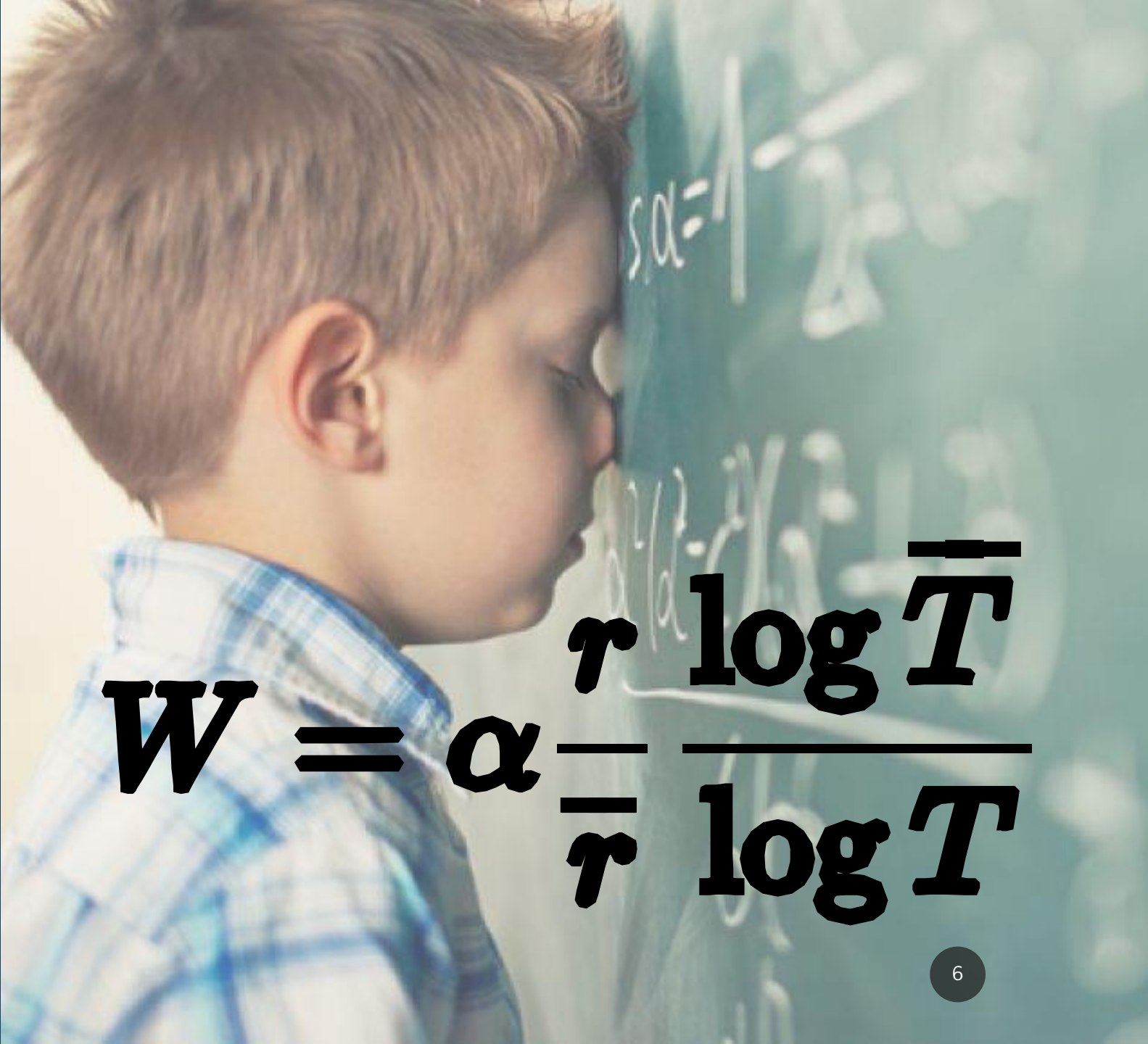


PERTE OU SANS PERTE

- Le JPEG est un algorithme de compression avec perte. Il permet de réduire à 1/10 de la taille initiale sans modification visible de la qualité.
- Le PNG, le FLAC sont des algorithmes sans perte, donc fichier plus lourd.

TAUX D'EFFICACITÉ DE COMPRESSION

- Le « score de Weissman », depuis la série est en train de devenir une réalité... ?
- C'était une méthode fictive universelle et absolue pour quantifier la puissance d'un algorithme de compression.


$$W = \alpha \frac{r \log \bar{T}}{\bar{r} \log T}$$

EFFICACITÉ DE LA COMPRESSION

Plus simplement utilisé aujourd'hui :

$$\sigma = \frac{\text{nombre de bits fichier original}}{\text{nombre de bits fichier compressé}}$$

ENTROPIE ET INFORMATION

- Claude Shannon a bâti sa **théorie de l'information** sur un concept nommé **entropie** (« quantité d'information »).



COMPRENDRE L'ENTROPIE

La chaîne de caractères
« **aaaaaaaaaaaaaaaaaaaa** » a
un taux d'entropie très
faible, car elle se résume à
« **16 fois a** ».

Au contraire,
« **1f08c6e13b954f80** » a un
taux d'entropie élevé, car
chaque caractère est
complètement aléatoire.

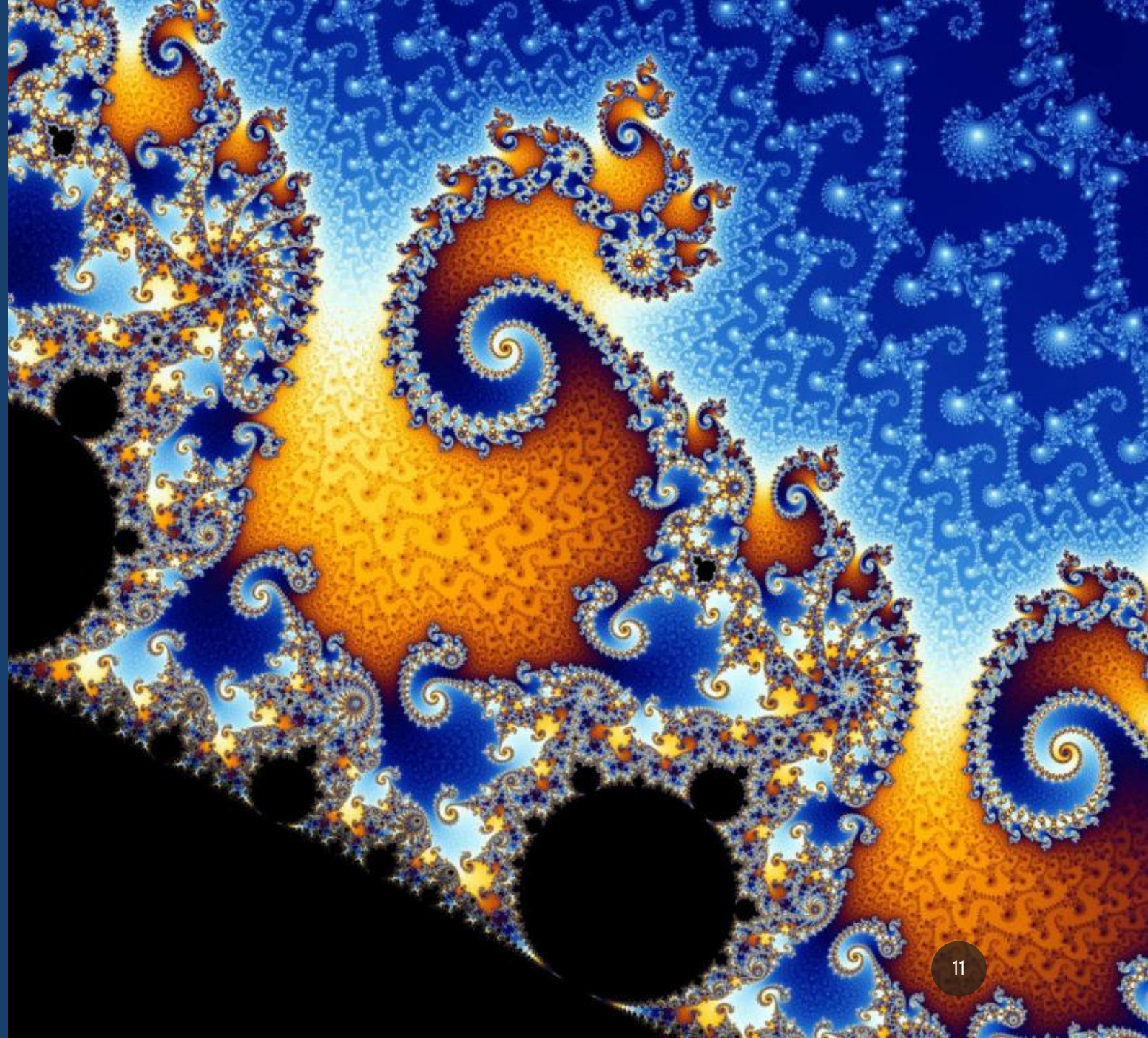
COMPRENDRE L'ENTROPIE

Si la manière la plus courte d'écrire « **aaaaaaaaaaaaaaaaaaaa** » est « **16 fois a** », cette chaîne de caractère est, selon Kolmogorov, beaucoup moins complexe que « **1f08c6e13b954f80** », dont la manière la plus courte de l'écrire est... « **1f08c6e13b954f80** ».



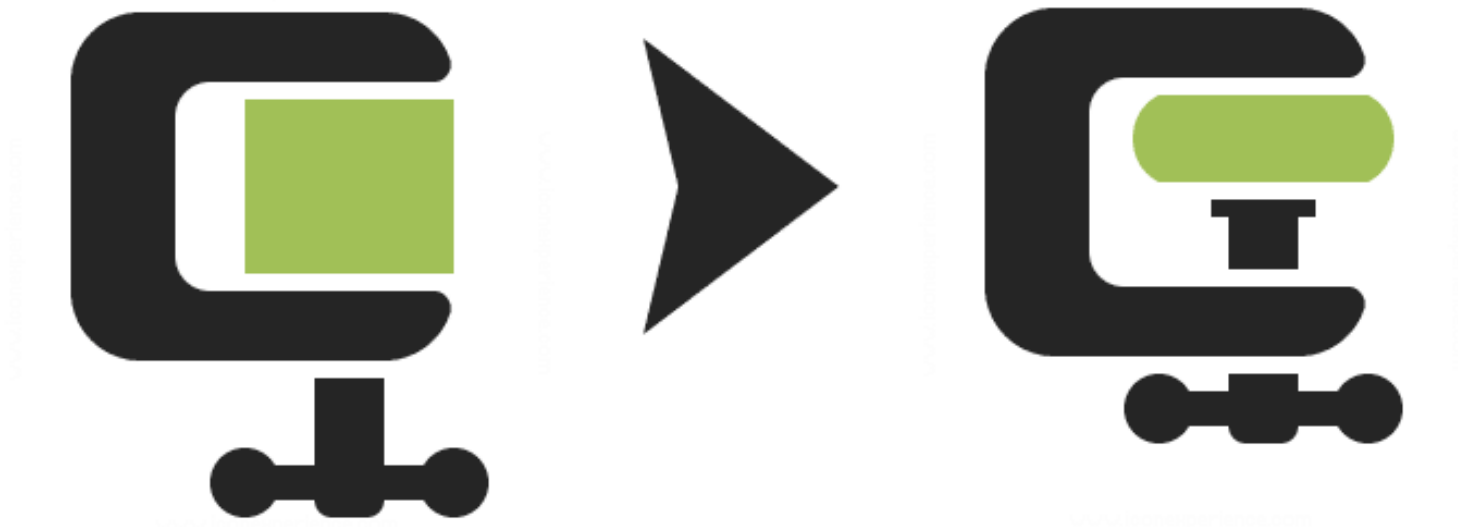
COMPRENDRE L'ENTROPIE

- En physique, entropie = quantité de désordre.
- Chaîne de caractères aléatoire, par définition complètement désordonnée, est celle ayant le plus d'entropie. Quand il y a peu d'entropie, c'est qu'il y a de l'ordre et de la logique, sous forme de motifs prévisibles.
- La fractale de Mandelbrot a une complexité de Kolmogorov faible car elle se résume à une équation de niveau Terminal S.



COMPRESSION

- Compresser un fichier, ça veut dire l'écrire de la manière la plus courte possible.
- On ne peut généralement pas regarder un fichier compressé et dire sans se tromper s'il peut ou non être encore plus compressé.



COMPRESSION

Exemple : Trouver « *le plus petit entier positif non définissable en moins de treize mots* ».

COMPRESSION

Exemple : Trouver « *le plus petit entier positif non définissable en moins de treize mots* ».

« *Le₁ plus₂ petit₃ entier₄ positif₅ non₆ définissable₇ en₈ moins₉ de₁₀ treize₁₁ mots₁₂* » vient d'être défini en douze mots. Monsieur Kolmogorov le comprimerait en combien de mots, du coup ? Douze ou au moins treize ?

COMPRESSION

La compression, on en fait tous les jours sans le savoir.

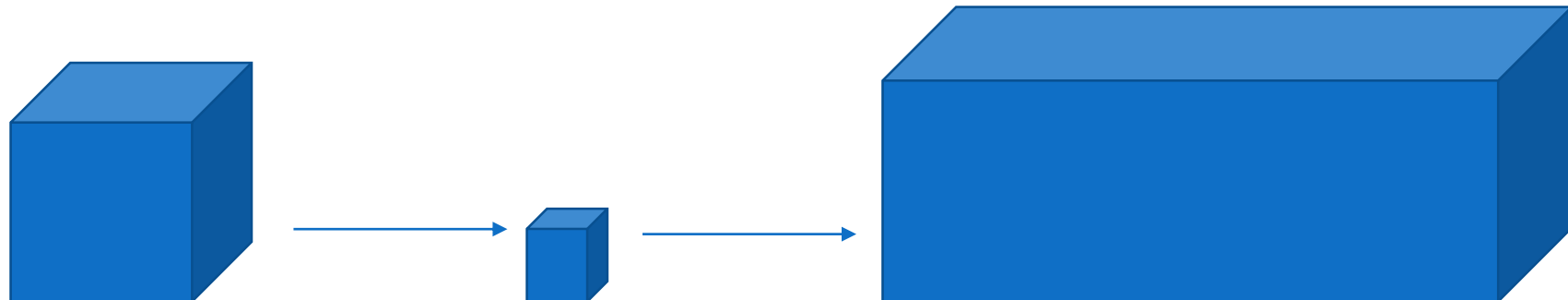


COMPRESSION RLE

- Le run-length encoding (**RLE**) est le plus basique.
- Séquence « aaaaaaaaaaaaaaaaaa » peut s'écrire « **16a** » **et on gagne treize caractères de place.**
- « 50 pixels noirs » au lieu d'écrire « pixel noir » cinquante fois de suite.
- « *UE* » à la place de « *Union européenne* », « *NASA* » pour « *National Aeronautics and Space Administration* » ou « *MP3* » pour « *MPEG-2 Audio Layer III* »
- ...

COMPRESSION

- Même le meilleur algorithme du monde ne pourra pas *tout compresser*.
- Compresser un fichier deux fois de suite donne, en général, un résultat plus volumineux que le fichier de départ.



COMPRESSION

- CODEC = Codeur
Décodeur.

Permet l'encodage ou le décodage d'un flux de données numérique, en vue d'une transmission ou d'un stockage.

- Algorithme ZIP marche relativement bien pour toutes sortes de données.
- Pour une meilleure compression, algorithme dédié selon le type de données.

LEMPER-ZIV-WELCH (ZIP)

```
FONCTION LZW_Compresser(Texte, dictionnaire)
```

```
  w ← ""
```

```
  TANT QUE (il reste des caractères à lire dans Texte) FAIRE
```

```
    c ← Lire(Texte)
```

```
    p ← Concaténer(w, c)
```

```
    SI Existe(p, dictionnaire) ALORS
```

```
      w ← p
```

```
    SINON
```

```
      Ajouter(p, dictionnaire)
```

```
      Écrire dictionnaire[w]
```

```
      w ← c
```

```
  FIN TANT QUE
```

LEMPER-ZIV- WELCH

- Créons le dictionnaire en y ajoutant les occurrences présentes dans le texte à compresser.
- $w \leftarrow \langle \rangle$
- $c \leftarrow \langle a \rangle$
- $p \leftarrow \langle \rangle a$
- $w \leftarrow a$

texte à compresser : a b a b c b a b a b a a a a a a

dictionnaire	a	b	c			. . .
	0	1	2	3	4	

LEMPER-ZIV- WELCH

- ...w <- a
- c <- b
- p <- ab
- ab ne se trouve pas encore dans le dictionnaire. Nous l'ajoutons.
- Nous ajoutons l'index de w au résultat (0).
- w <- b

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0

a	b	c	ab		. . .
0	1	2	3	4	

- Mot ajouté au dictionnaire
- Prefixe du mot ajouté au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

- ...w <- b
- c <- a
- p <- ba
- ba ne se trouve pas encore dans le dictionnaire. Nous l'ajoutons.
- Nous ajoutons l'index de w au résultat (1).
- w <- a

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1

a	b	c	ab	ba	. . .
0	1	2	3	4	

- Mot ajouté au dictionnaire
- Prefixe du mot ajouté au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

- ...W ← a
- c ← b
- p ← ab
- ab se trouve déjà dans le dictionnaire.
- w ← b

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3

a	b	c	ab	ba	abc	
0	1	2	3	4	5	

- Mot ajouté au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

- Et ainsi de suite...

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3__2

a	b	c	ab	ba	abc	cb		
0	1	2	3	4	5	6		

- Mots ajoutés au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3__2_4

a	b	c	ab	ba	abc	cb	bab	
0	1	2	3	4	5	6	7	

- Mots ajoutés au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3__2_4__7

a	b	c	ab	ba	abc	cb	bab	baba	
0	1	2	3	4	5	6	7	8	

- Mots ajoutés au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3__2_4__7____0

a	b	c	ab	ba	abc	cb	bab	babaaa	
0	1	2	3	4	5	6	7	8	9

- Mots ajoutés au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3__2_4__7____0_9

a	b	c	ab	ba	abc	cb	bab	babaaa	aaa	
0	1	2	3	4	5	6	7	8	9	10

- Mots ajoutés au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

texte à compresser : a b a b c b a b a b a a a a a a a

resultat 0_1_3__2_4__7____0_9__10

a	b	c	ab	ba	abc	cb	bab	babaaa	aaa	aaaa	
0	1	2	3	4	5	6	7	8	9	10	11

- Mots ajoutés au dictionnaire
- Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPER-ZIV- WELCH

- Finalement nous avons remplacé 17 caractères par 10 nombres.

texte à compresser : a b a b c b a b a b a a a a a a

resultat 0_1_3__2_4__7____0_9__10____0

a	b	c	ab	ba	abc	cb	bab	babaaa	aaa	aaaa	
0	1	2	3	4	5	6	7	8	9	10	11

■ Mots ajoutés au dictionnaire

■ Prefixes des mots ajoutés au dictionnaire(le mot sans son dernier caractère)

LEMPERL-ZIV-WELCH (UNZIP)

```
FONCTION LZW_Décompresser(Code, dictionnaire)
```

```
  n ← |Code|
```

```
  v ← Lire(Code)
```

```
  Écrire dictionnaire[v]
```

```
  w ← chr(v)
```

```
  POUR i ALLANT DE 2 à n FAIRE
```

```
    v ← Lire(Code)
```

```
    SI Existe(v, dictionnaire) ALORS
```

```
      entrée ← dictionnaire[v]
```

```
    SINON entrée ← Concaténer(w, w[0])
```

```
      Écrire entrée
```

```
      Ajouter(Concaténer(w,entrée[0]), dictionnaire)
```

```
      w ← entrée
```

```
  FIN POUR
```


LEMPER-ZIV- WELCH

- On sait que 0 correspond à la première lettre de l'alphabet.

code à décompresser : 0 1_3_2_4_7_0_9_10_0

resultat a

dictionnaire	a	b	c	
	0	1	2	

- Mots ajoutés au dictionnaire
- Code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- On connaît la valeur pour l'indice 1.
- On suppose donc que ab avait été ajouté au dictionnaire.

code à décompresser : 0 1 3 2 4 7 0 9 10 0

resultat a b

dictionnaire	a	b	c	ab	
	0	1	2	3	

- Mots ajoutés au dictionnaire
- Code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV-WELCH

- L'indice 3 est déjà présent dans le dictionnaire.
- On suppose que ba a donc été ajouté.

code à décompresser : 0 1 3__ 2_4__7__0_9__10__0

resultat a b a b

dictionnaire	a	b	c	ab	ba	
	0	1	2	3	4	

- Mots ajoutés au dictionnaire
- Code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 2 existe dans le dictionnaire.
- Nous avons ab (en rouge), nous supposons que abc a donc été ajouté dans le dictionnaire.

code à décompresser : 0_1 3__ 2 4__7__0_9__10__0

resultat a b a b c

dictionnaire	a	b	c	ab	ba	abc	
	0	1	2	3	4	5	

- Mots ajoutés au dictionnaire
- Code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 4 existe dans le dictionnaire.
- Nous supposons que cb a donc été ajouté dans le dictionnaire.

code à décompresser : 0_1_3_ 2 4 7_0_9_10_0

resultat a b a b c b a

dictionnaire	a	b	c	ab	ba	abc	cb	
	0	1	2	3	4	5	6	

- Mots ajoutés au dictionnaire
- Code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 7 n'existe pas encore dans le dictionnaire. Nous l'ajoutons en concaténant le code décodé avant (en rouge) avec sa première lettre (b).
- Nous supposons que bab ainsi entré correspond au code de cette valeur.

code à décompresser : 0_1_3__2 4 7 0_9__10__0

resultat a b a b c b a b a b

dictionnaire	a	b	c	ab	ba	abc	cb	bab	
	0	1	2	3	4	5	6	7	

- Mots ajoutés au dictionnaire
- code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 0 existe dans le dictionnaire.
- Nous supposons que baba a donc été ajouté dans le dictionnaire.

code à décompresser : 0_1_3__2_4_ 7_____ 0_ 9__10____0

resultat a b a b c b a b a b a

dictionnaire	a	b	c	ab	ba	abc	cb	bab	baba	
	0	1	2	3	4	5	6	7	8	

- Mots ajoutés au dictionnaire
- code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 9 n'existe pas encore dans le dictionnaire. Nous l'ajoutons en concaténant le code décodé avant (en rouge) avec sa première lettre (a).
- Nous supposons que aa ainsi entré correspond au code de cette valeur.

code à décompresser : 0_1_3__2_4_7_____ 0 9__ 10____0

resultat a b a b c b a b a b a a a

dictionnaire	a	b	c	ab	ba	abc	cb	bab	babaa	
	0	1	2	3	4	5	6	7	8	9

- Mots ajoutés au dictionnaire
- code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 10 n'existe pas encore dans le dictionnaire. Nous l'ajoutons en concaténant le code décodé avant (en rouge) avec sa première lettre (a).
- Nous supposons que aaa ainsi entré correspond au code de cette valeur.

code à décompresser : 0_1_3__2_4_7____0 9__ 10__ 0

resultat a b a b c b a b a b a a a a a a

dictionnaire	a	b	c	ab	ba	abc	cb	bab	babaaa	aaa	
	0	1	2	3	4	5	6	7	8	9	10

- Mots ajoutés au dictionnaire
- code en cours de décompression
- Code décompressé à l'itération d'avant

LEMPER-ZIV- WELCH

- 0 existe dans le dictionnaire.
- Nous supposons que aaaa a donc été ajouté dans le dictionnaire.

code à décompresser : 0_1_3__2_4_7____0_ 9_ 10__ 0

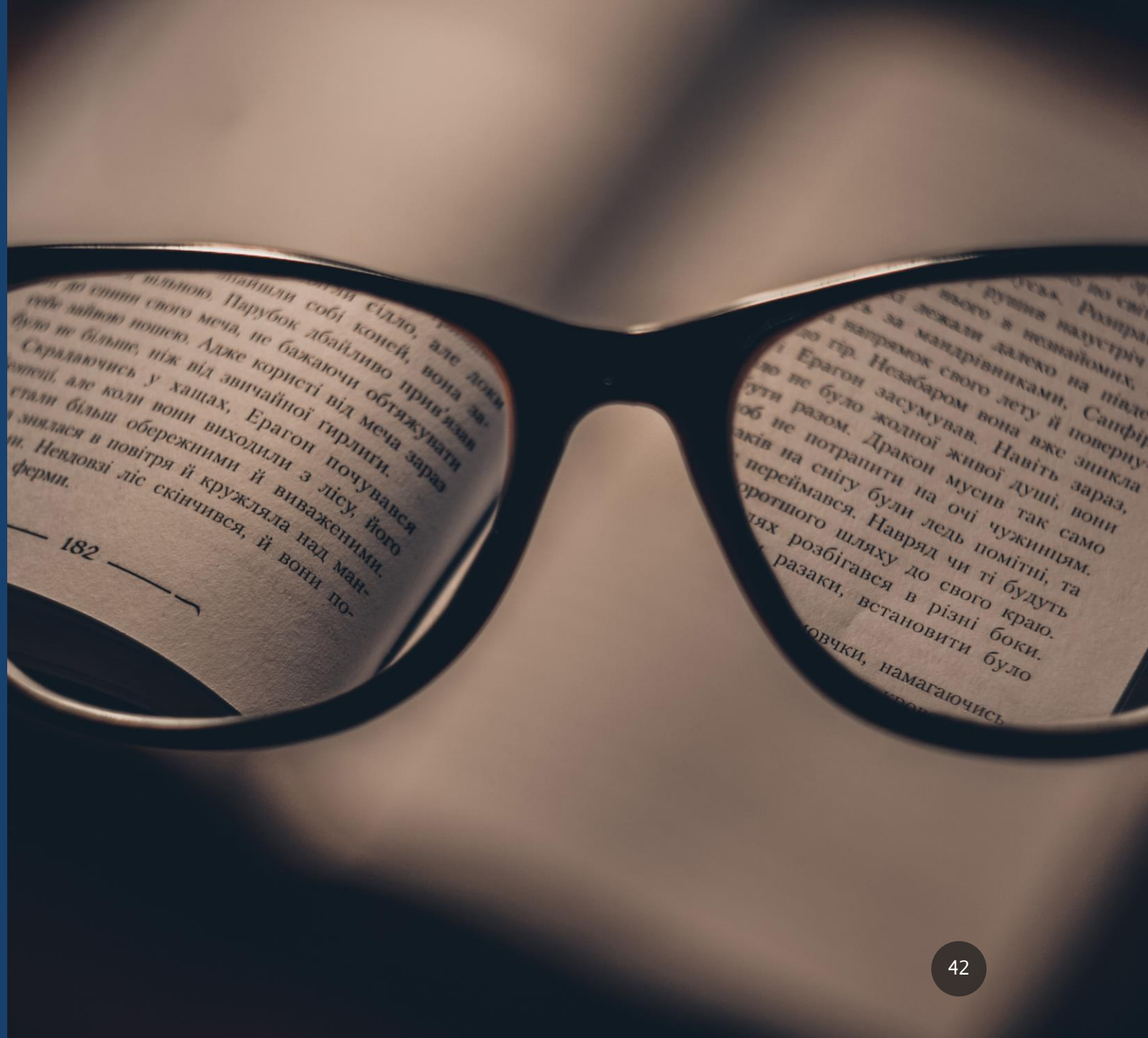
resultat a b a b c b a b a b a a a a a a a

dictionnaire	a	b	c	ab	ba	abc	cb	bab	babaaa	aaa	aaaa	
	0	1	2	3	4	5	6	7	8	9	10	11

- Mots ajoutés au dictionnaire
- code en cours de décompression
- Code décompressé à l'itération d'avant

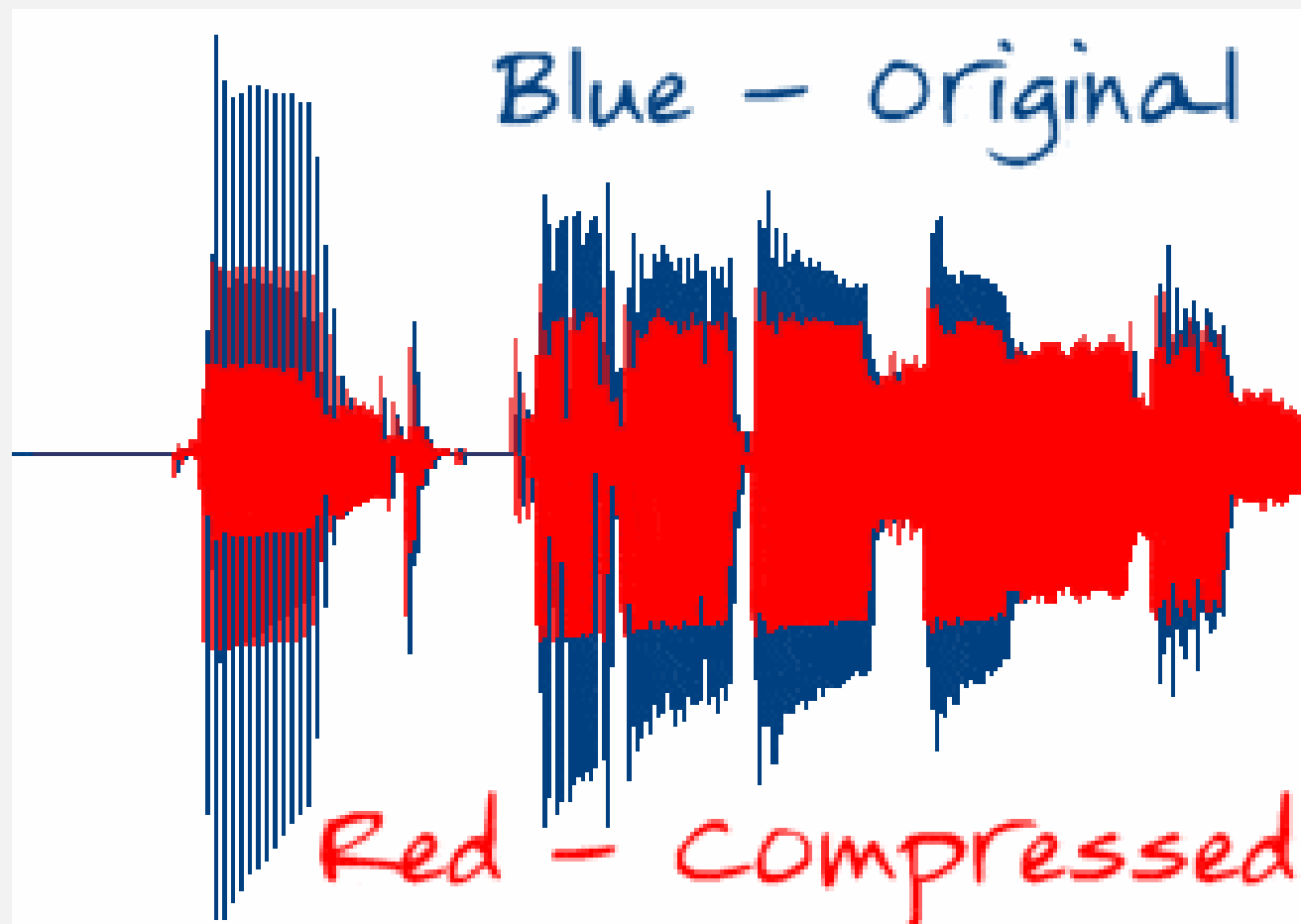
LEMPEL- ZIV- WELCH

Pour information, la langue anglaise contient 75% de redondance.

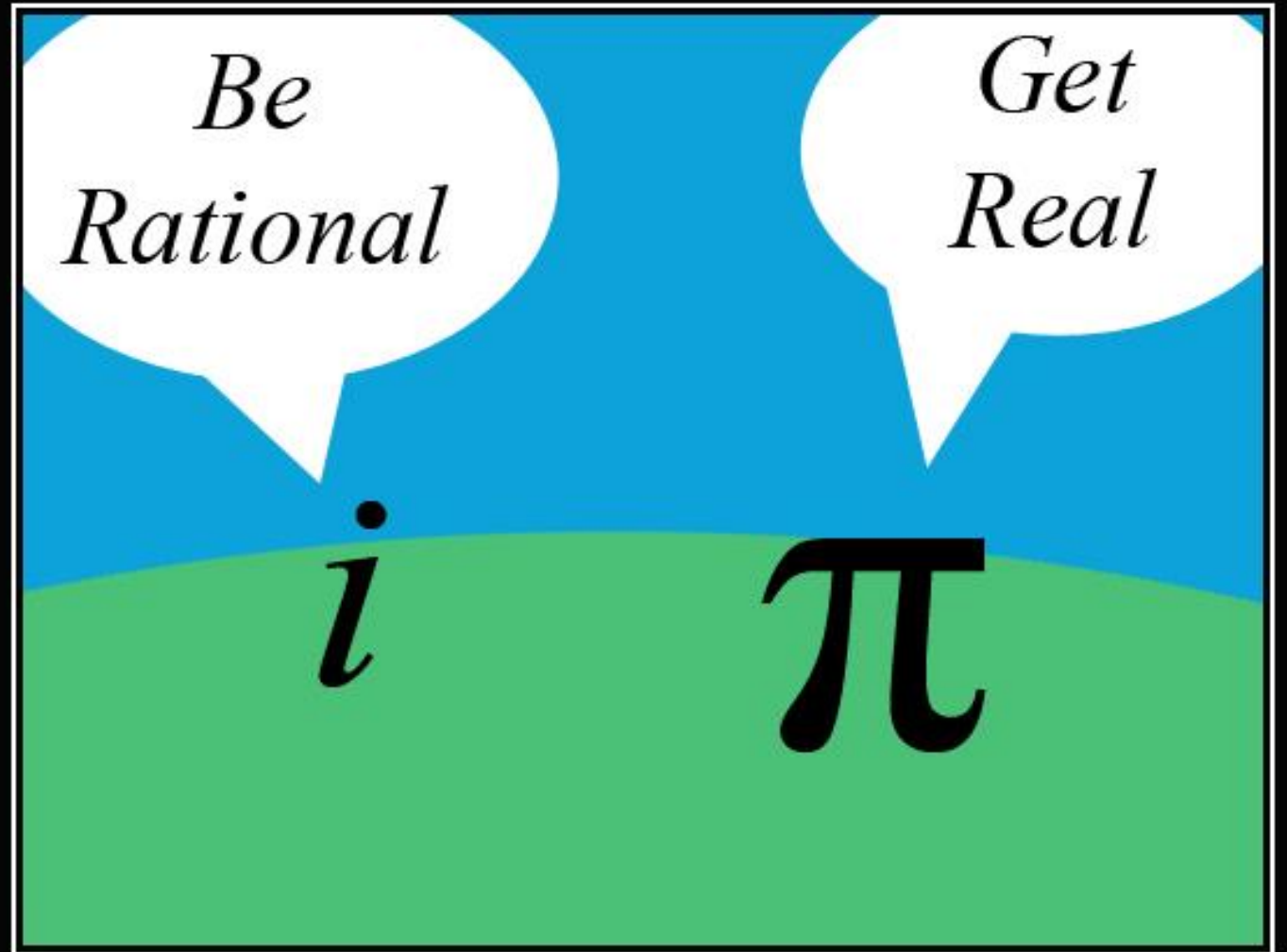


COMPRESSION AUDIO

Peut permettre
également
d'embellir le son
selon l'effet
recherché...



J'AI HATE DE
CODER MON
CODEC !!!



MATH JOKES

If you get them, you probably don't have any friends.