

Ceci est une planche écrite avec la classe Beamer en \LaTeX .
Par défaut, la gestion des paragraphes (espacement, retrait) est
différente des articles.

On peut aussi
faire des boîtes.

Ce document est destiné à être compilé avec Lua \LaTeX ou Xe \LaTeX ,
une des implémentations modernes de \LaTeX :

Ce document est destiné à être compilé avec Lua \LaTeX ou Xe \LaTeX , une des implémentations modernes de \LaTeX :

Xe \LaTeX moteur développé en 2004

Lua \LaTeX développé à partir de 2007, il est un peu plus lent que Xe \LaTeX mais ajoute le support du langage de script Lua, ce qui autorise des fonctionnalités avancées (placement automatique des sommets d'un graphe avec TikZ, par exemple).

Ce document est destiné à être compilé avec Lua \LaTeX ou Xe \LaTeX , une des implémentations modernes de \LaTeX :

Xe \LaTeX moteur développé en 2004

Lua \LaTeX développé à partir de 2007, il est un peu plus lent que Xe \LaTeX mais ajoute le support du langage de script Lua, ce qui autorise des fonctionnalités avancées (placement automatique des sommets d'un graphe avec TikZ, par exemple).

Remarque

- ▶ pdf \LaTeX , le moteur créé à partir des années 1990, n'est plus développé et est voué à l'obsolescence ;
- ▶ \LaTeX , le moteur historique, est obsolète — si vous voyez une source vous parlant de compiler du \LaTeX en DVI, méfiez-vous : c'est très vieux et obsolète.

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, `show` est affiché à partir du calque 2 ; `hide` est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : ce texte va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : ce texte va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; celui-ci est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : ce texte va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : ce texte va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : ce texte va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi **mettre en valeur** du texte sur certains calques. On peut

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut **combiner les deux**.

Il est également possible d'utiliser des commandes pour masquer du texte sur certains calques.

Par exemple, ce texte est affiché à partir du calque 2 ; est affiché seulement sur le calque 3.

Il existe une commande qui fait plus que cacher le texte : va disparaître et libérer sa place au calque 6. Pour du texte, ce n'est pas terrible, mais pour des figures, c'est utile.

On peut aussi mettre en valeur du texte sur certains calques. On peut combiner les deux.

Mélanger les pauses et ces dispositifs n'est pas aisé, mais l'argument facultatif de la commande permet de recoller les morceaux.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification `<+-->` a juste été appliquée à chaque item.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle + - \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle + - \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle + - \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation .

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle + - \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation n'importe où.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle + - \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation n'importe où.
6. Il existe aussi une notation $.$, qui prend
mais $.$

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle ++ \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation n'importe où.
6. Il existe aussi une notation $.$, qui prend la valeur du compteur de pause moins 1 mais sans l'incrémenter.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle ++ \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation n'importe où.
6. Il existe aussi une notation $.$, qui prend la valeur du compteur de pause moins 1 mais sans l'incrémenter.
7. Ça permet d'afficher plusieurs bouts en même temps.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification $\langle ++ \rangle$ a juste été appliquée à chaque item.
3. Le $+$ signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le $-$ ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation n'importe où.
6. Il existe aussi une notation $.$, qui prend la valeur du compteur de pause moins 1 mais sans l'incrémenter.
7. Ça permet d'afficher plusieurs bouts en même temps.

Afficher une liste progressivement

On peut utiliser un argument supplémentaire de l'environnement de liste.

1. Chaque item s'affiche l'un après l'autre.
2. En réalité, la spécification `<+-->` a juste été appliquée à chaque item.
3. Le `+` signifie : utiliser la valeur du compteur de pause, puis incrémenter ce compteur.
4. Le `-` ensuite signifie comme d'habitude que ça reste affiché après ce moment.
5. On peut utiliser cette notation n'importe où.
6. Il existe aussi une notation `.`, qui prend la valeur du compteur de pause moins 1 mais sans l'incrémenter.
7. Ça permet d'afficher plusieurs bouts en même temps.

Il y a un décalage quand on fait une pause après une énumération.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

- ▶ Chaque item apparait l'un après l'autre.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

- ▶ Chaque item apparait l'un après l'autre.
- ▶ Le `+` fait la même chose que tout à l'heure, mais le numéro de calque utilisé pour l'affichage de chaque item est augmenté de 1.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

- ▶ Chaque item apparaît l'un après l'autre.
- ▶ Le `+` fait la même chose que tout à l'heure, mais le numéro de calque utilisé pour l'affichage de chaque item est augmenté de 1.
- ▶ C'est pour ça que l'énumération commence à apparaître en calque 2, bien qu'il n'y ait pas de pause avant.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

- ▶ Chaque item apparaît l'un après l'autre.
- ▶ Le `+` fait la même chose que tout à l'heure, mais le numéro de calque utilisé pour l'affichage de chaque item est augmenté de 1.
- ▶ C'est pour ça que l'énumération commence à apparaître en calque 2, bien qu'il n'y ait pas de pause avant.

Et pas de décalage non plus, car il n'y a pas de (1) dans la commande pause. On peut **n'importe où**.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

- ▶ Chaque item apparaît l'un après l'autre.
- ▶ Le `+` fait la même chose que tout à l'heure, mais le numéro de calque utilisé pour l'affichage de chaque item est augmenté de 1.
- ▶ C'est pour ça que l'énumération commence à apparaître en calque 2, bien qu'il n'y ait pas de pause avant.

Et pas de décalage non plus, car il n'y a pas de (1) dans la commande pause. On peut utiliser `pause` n'importe où.

Afficher une liste progressivement

Autre méthode

On peut décaler le résultat de `.` ou `+` d'une valeur : ça évite de devoir mettre une pause avant l'énumération et ça élimine le décalage si on met une pause après.

- ▶ Chaque item apparaît l'un après l'autre.
- ▶ Le `+` fait la même chose que tout à l'heure, mais le numéro de calque utilisé pour l'affichage de chaque item est augmenté de 1.
- ▶ C'est pour ça que l'énumération commence à apparaître en calque 2, bien qu'il n'y ait pas de pause avant.

Et pas de décalage non plus, car il n'y a pas de (1) dans la commande pause. On peut utiliser cela n'importe où.

Afficher du code

Il est d'usage d'utiliser une police à chasse fixe. Mais ça ne suffit pas : si notre code contient des `\` ou autres caractères ayant un sens spécial en \LaTeX , c'est un peu pénible.

Afficher du code

Il est d'usage d'utiliser une police à chasse fixe. Mais ça ne suffit pas : si notre code contient des `\` ou autres caractères ayant un sens spécial en \LaTeX , c'est un peu pénible.

La commande `\verb` permet de désactiver temporairement le sens particulier de `\`. Pour cela, elle modifie temporairement ce qu'on appelle des *catcodes*. Quand on le fait avec beamer, cela pose des problèmes supplémentaires, car par défaut, beamer lit tout le code d'une planche d'un coup pour calculer quels calques créer, et ce faisant les modifications de catcodes n'ont pas lieu au bon moment.

Afficher du code

Il est d'usage d'utiliser une police à chasse fixe. Mais ça ne suffit pas : si notre code contient des `\` ou autres caractères ayant un sens spécial en \LaTeX , c'est un peu pénible.

La commande `\verb` permet de désactiver temporairement le sens particulier de `\`. Pour cela, elle modifie temporairement ce qu'on appelle des *catcodes*. Quand on le fait avec beamer, cela pose des problèmes supplémentaires, car par défaut, beamer lit tout le code d'une planche d'un coup pour calculer quels calques créer, et ce faisant les modifications de catcodes n'ont pas lieu au bon moment.

Pour pallier ça, on ajoute l'option `[fragile]` au début : beamer écrit alors le code de la planche dans un fichier temporaire, et le relit autant de fois que nécessaire (ce processus force \LaTeX à évaluer à nouveau les catcodes).

Des paquets comme Listings permettent d'enjoliver la présentation d'un code source, en fonction de la syntaxe.

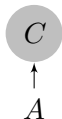
```
def f(x) :  
    """une chaine de doc"""  
    return x # pas très utile
```

Des paquets comme Listings permettent d'enjoliver la présentation d'un code source, en fonction de la syntaxe.

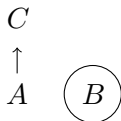
```
def f(x) :  
    """une chaine de doc"""  
    return x # pas très utile
```

L'inconvénient, c'est qu'on ne peut pas facilement animer ces codes. D'autres solutions existent :
chato-beamer-codesource.sty fournit un environnement dans lequel certains caractères spéciaux sont désactivés, mais pas \, ce qui permet d'y utiliser \uncover par exemple.

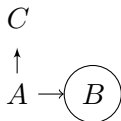
Pour faire des figures, on utilise TikZ, créé par la même personne que Beamer : les deux sont très compatibles car ils reposent sur le même paquet technique, PGF.



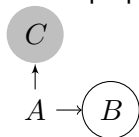
Pour faire des figures, on utilise TikZ, créé par la même personne que Beamer : les deux sont très compatibles car ils reposent sur le même paquet technique, PGF.



Pour faire des figures, on utilise TikZ, créé par la même personne que Beamer : les deux sont très compatibles car ils reposent sur le même paquet technique, PGF.



Pour faire des figures, on utilise TikZ, créé par la même personne que Beamer : les deux sont très compatibles car ils reposent sur le même paquet technique, PGF.



Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tabluarx, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tabluarx, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

a	

Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tabluarx, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

<i>a</i>	<i>b</i>

Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tabluarx, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

<i>a</i>	<i>b</i>
<i>d</i>	

Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tabluarx, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

<i>a</i>	<i>b</i>
<i>d</i>	e_1

Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tabluarx, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

<i>a</i>	<i>b</i>
<i>d</i>	e_1
<i>f</i>	

Pour faire des tableaux, il y a des environnements disponibles de base (array, tabular) et de nombreux paquets qui en ajoutent de nouveaux (tbluax, longtable, etc.).

Je recommande tabularray, qui reprend et améliore les fonctionnalités de nombreux paquets ; c'est indéniablement l'avenir pour la création de tableaux.

a	b
d	e_1
f	$\sin(x)$

C'est pratique pour les abréviations d'utiliser une commande.

Attentions aux espaces : \LaTeX les ignore après le nom d'une commande, ce pourquoi on ajoute un $\{ \}$.

Une fonction de classe \mathcal{C}^∞ , ou \mathcal{C}^1 , ou \mathcal{C}^n , ou \mathcal{C}^{n+1} .

Un essai obligatoire : fructueux ; optionnel : défaut ; un autre obligatoire : plus ; optionnel : complet. Notez que le paramètre entre crochets a été pris en compte malgré l'espace qui le précède ; si on avait défini la commande avec $\mathfrak{m}!o$, ça n'aurait pas été le cas.

Voir la doc de `xparse`.

On n'a rien mis après `machin`

Pour plus d'informations pour utiliser :

- ▶ <https://www.ctan.org/> n'est pas très pratique mais c'est là qu'on trouve les documentations des paquets.
- ▶ <https://tex.stackexchange.com/> est un bon site de questions/réponses. Mais regarder la date des messages : il y a aussi des vieilleries qui traînent.
- ▶ Éviter la plupart des autres forums en ligne, qui colportent des réponses de qualité très inégale, et souvent obsolètes, et surtout les sites d'apparence vieillotte : le contenu a probablement l'âge de la présentation.

Plus profondément :

- ▶ <https://www.ctan.org/pkg/texbytopic> explique le fonctionnement interne de T_EX.
- ▶ <https://www.ctan.org/pkg/luatex> explique spécifiquement le moteur de LuaT_EX et l'interaction avec Lua (un peu indigeste).
- ▶ <https://www.latex-project.org/> pour les informations sur les développements modernes de L^AT_EX.
- ▶ <http://mirrors.ctan.org/info/l2tabu/french/l2tabufr.pdf> Recense les mauvaises pratiques, notamment les commandes obsolètes depuis L^AT_EX 2_ε (1994 !); attention, certaines recommandations sont à leur tour devenues obsolètes, mais c'est une bonne ressource à consulter si on récupère du code vieux.