

# BNF - scenario

```
<deck> ::= <deck_frontmatter>? <slide> (<slide_sep> <slide>)*

<slide_sep> ::= NL '---' NL

; -----
; DECK (metadata globale)
; -----
<deck_frontmatter> ::= '---' NL <deck_meta_line>* '---' NL

<deck_meta_line> ::=
    'title:' WS <string> NL
    | 'author:' WS <string> NL
    | 'offlineExport:' WS <bool> NL
    | 'template:' WS <string> NL
    | 'cssPath:' WS <path> NL
    | 'assetsDir:' WS <path> NL

; -----
; SLIDE
; -----
<slide> ::= <slide_frontmatter>? <slide_body>

<slide_frontmatter> ::= '---' NL <slide_meta_line>* '---' NL

<slide_meta_line> ::=
    'id:' WS <id> NL
    | 'layout:' WS <layout_name> NL
    | 'transition:' WS <transition_type> (WS <int>)? NL
    | 'notes:' WS '||' NL <indented_lines>

<layout_name> ::= <id>           ; ex: TITLE_CONTENT, TWO_COLUMNS, FREE (ou nom custom)
<transition_type> ::= 'NONE' | 'FADE' | 'SLIDE' | 'ZOOM'

<slide_body> ::= (<block> NL?)*

; -----
; BLOCKS (contenu)
; Chaque block peut être "révélé" à un step avec le préfixe @n
; -----
<block> ::= <reveal_prefix>? (
```

```

        <heading>
    | <paragraph>
    | <list>
    | <image>
    | <video>
    | <code_block>
    | <equation_block>
    | <zone_block>
    | <slot_block>
    )

<reveal_prefix> ::= '@' <int> WS ; apparaît à l'étape <int>

; -----
; TEXTE / LISTES
; -----
<heading> ::= ('#' | '##' | '###') WS <textline>

<paragraph> ::= <textline> (NL <textline>)*

<list> ::= <unordered_list> | <ordered_list>
<unordered_list> ::= ('-' WS <textline> NL)+
<ordered_list> ::= (<int> '.' WS <textline> NL)+

; -----
; MEDIA
; -----
<image> ::= '!' '[' <alt_text> ']' '(' <path> ')' <image_opts>?
<image_opts> ::= WS '{' 'fit=' <image_fit> '}'
<image_fit> ::= 'CONTAIN' | 'COVER' | 'STRETCH'

<video> ::= '!video' '(' <path> ')' <video_opts>?
<video_opts> ::= WS '{' <video_flag> (WS <video_flag>)* '}'
<video_flag> ::= 'autoplay' | 'loop' | 'muted' | 'controls'

; -----
; CODE + REVEAL LIGNES
; {1|2-3|4} => étapes successives de révélation des lignes
; -----
<code_block> ::= '```' <lang> <code_reveal_spec>? NL <code_lines> ````
<lang> ::= <id>

<code_reveal_spec> ::= WS '{' <reveal_range> ('|' <reveal_range>)* '}''
<reveal_range> ::= <int> | <int> '-' <int>

<code_lines> ::= (<textline> NL)*

```

```

; -----
; EQUATIONS (extension)
; -----
<equation_block> ::= <equation_inline> | <equation_display>
<equation_inline> ::= '$' <latex> '$'
<equation_display> ::= '$$' NL <latex_lines> '$$'

<latex_lines> ::= (<textline> NL)*

; -----
; ZONES (layout zones explicites)
; Exemple:
; ::zone LEFT::
;   @1 - item
; ::endzone::
; -----
<zone_block> ::= '::zone' WS <area> '::' NL <slide_body> '::endzone::'
<area> ::= 'HEADER' | 'CONTENT' | 'LEFT' | 'RIGHT' | 'FOOTER' | 'CANVAS'

; -----
; SLOTS + SWITCH (remplacement contrôlé)
; Un slot définit des variantes affichées selon le step.
; Exemple:
; ::slot viz
;   @1 ![a](a.png)
;   @2 ![b](b.png)
; :::
; -----
<slot_block> ::= '::::slot' WS <slot_id> NL <slot_entry>+ '::::'
<slot_id> ::= <id>

<slot_entry> ::= '@' <int> WS <slot_variant_block> NL?
<slot_variant_block> ::= <image> | <video> | <paragraph> | <zone_block>

; -----
; LEXICAL
; -----
<bool> ::= 'true' | 'false'
<id> ::= <letter> (<letter> | <digit> | '_' | '-')*
<int> ::= <digit>+
<path> ::= <string>
<string> ::= "" <chars>* "" | <bare_string>
<bare_string> ::= <chars_no_nl>+
<alt_text> ::= <chars_no_bracket>*
<textline> ::= <chars_no_nl>+

```

```
<indented_lines> ::= (WS WS <textline> NL)*

<letter> ::= 'A'..'Z' | 'a'..'z'
<digit> ::= '0'..'9'
<latex> ::= <chars_no_dollar>+
<chars>* ::= (any char)*
WS ::= ' '
NL ::= '\n'
```

```
---
title: "Apprenticeship Slides"
author: "Student"
offlineExport: true
template: "company-template"
cssPath: "themes/company/style.css"
assetsDir: "public/"
---
```

```
---
id: title
layout: TITLE_CONTENT
transition: FADE 500
---
```

```
# SophiaTech Eats
@1 Current status: In progress
@2 ![Workplace](images/site.jpg)
```

```
---
id: outline
layout: TITLE_CONTENT
transition: SLIDE 400
---
```

```
# Outline
@1 - Context
@2 - Objectives
@3 - Architecture
@4 - Results
@5 - Conclusion
```

```
id: animations
layout: TWO_COLUMNS
transition: NONE
---

# Work done

::zone LEFT::
@1 I set up the CI/CD pipeline.
@2 I added caching to reduce build time.
@3 I improved integration tests.
::endzone::

::zone RIGHT::
@1 ! [Pipeline](images/pipeline.png)
@2 ! [Caching](images/cache.png)
::endzone::

---

id: demo
layout: FULLSCREEN_MEDIA
transition: ZOOM 400
---

# Demo in situ
@1 !video(videos/demo.mp4) {controls}

---

id: conclusion
layout: TITLE_CONTENT
transition: FADE 400
---

# Conclusion
@1 - Delivered features
@2 - Lessons learned
@3 - Next steps
```