

## # Block Reference

This manuscript renders every MVP block tag at 1:1 fidelity. Regenerate with ``renderer/scripts/regenerate-block-reference.sh`` after adding or changing a block.

> **Universal attrs:** Every block tag accepts ``width=N`` (1-100, percent of writeable text column, centred) and ``annotation=text`` (styled note below the block, ~85% body-size, italic, #666666). These aren't repeated in each block's parameter list.

---

## ## card

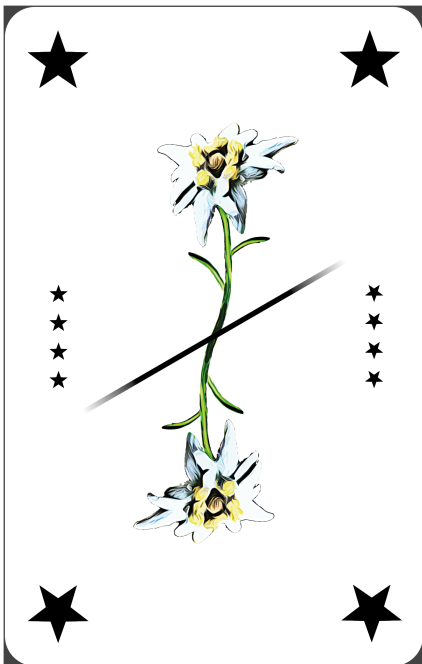
Renders a single card image at the default display size. Use ``deck`` and ``id`` to identify the card; the optional ``angle`` attribute rotates the image by the given number of degrees.

**Signature:** ``[card deck=<string> id=<card-id> angle=<int>]``

**Parameters:**

- ``deck`` (required, string) — Deck identifier (e.g. ``alpine-tarok``). - ``id`` (required, card-id) — Canonical card identifier within that deck (e.g. ``Skus``, ``t21``, ``hK``). - ``angle`` (optional, int, default 0) — Rotation in degrees, clockwise.

**Examples:**





---

## card-row

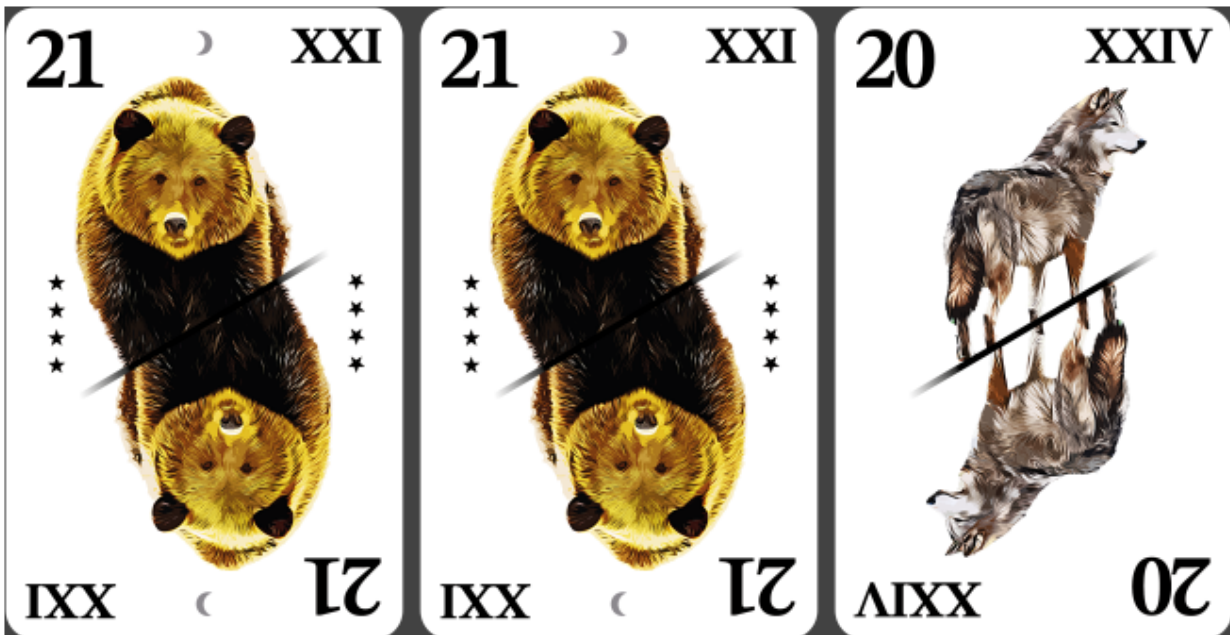
Renders a horizontal row of card images. Specify the cards as a comma-separated list of card-ids in `cards`. The optional `labels` attribute adds a caption beneath each card.

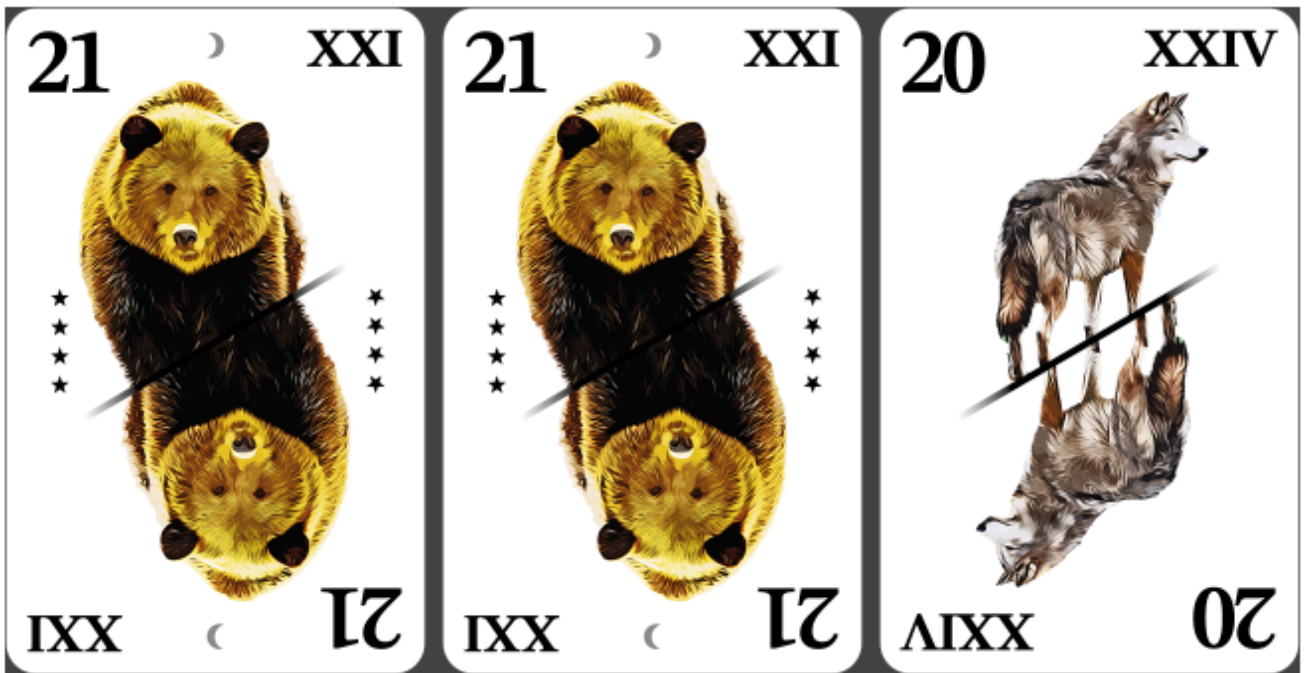
**\*\*Signature:\*\*** `[card-row deck=<string> cards=<comma-separated card-ids> labels=<string>]`

**\*\*Parameters:\*\***

- `deck` (required, string) — Deck identifier. - `cards` (required, comma-separated card-ids) — Ordered list of card ids to display left-to-right. - `labels` (optional, string) — Comma-separated labels aligned under each card.

**\*\*Examples:\*\***





---

## card-grid

Renders a grid of card images. Cards are laid out row by row from left to right. The number of columns is determined automatically based on the number of cards and the page width.

**Signature:** `[card-grid deck=<string> cards=<comma-separated card-ids>]`

**Parameters:**

- `deck` (required, string) — Deck identifier. - cards` (required, comma-separated card-ids) — Cards to display in the grid.`

**Examples:**

21 XXI  
\* \* \* \* \*  
\* \* \* \* \*  
IXX 21



21 XXI  
\* \* \* \* \*  
\* \* \* \* \*  
IXX 21



20 XXIV  
ΛΙΧΧ 20




19 XIX  
XIX 19



18 XVIII  
IIIAX 18



17 XVII  
IIAX 17





---

## card-1 — variant 1

Card-1 block — single-card showcase layout. The `variant` attribute selects the visual treatment (1 = full-width headline, 2 = side-by-side, 3 = compact). An optional `caption` appears below the card.

\*\*Signature:\*\* `[card-1 cards=<card-id> variant=<int 1|2|3> caption=<string>]`

\*\*Parameters:\*\*

- `cards` (required, card-id) — A single card id. - `variant` (optional, int 1|2|3, default 1) — Visual layout variant. - `caption` (optional, string) — Caption text displayed below the card.

\*\*Examples:\*\*



Card 1 — variant 1

---

## card-1 — variant 2



Card 1 — variant 2

---

## card-1 — variant 3



Card 1 — variant 3

---

## card-2 — variant 1

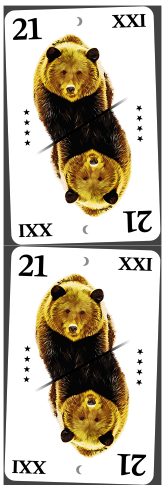
Card-2 block — two-card layout. Accepts exactly two card ids in `cards`. The `variant` attribute controls the presentation style.

**Signature:** `[card-2 cards=<comma-separated card-ids> variant=<int 1|2|3> caption=<string>]`

**Parameters:**

- `cards` (required, comma-separated card-ids) — Exactly two card ids. - `variant` (optional, int 1|2|3, default 1) — Visual layout variant. - `caption` (optional, string) — Caption text displayed below the cards.

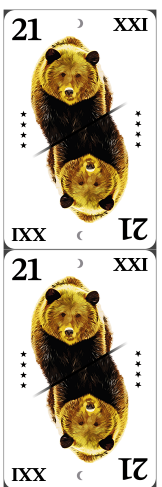
**Examples:**



Card 2 — variant 1

---

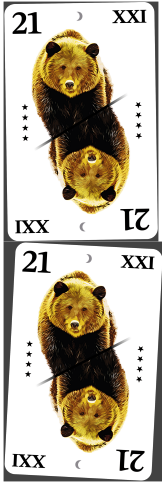
## card-2 — variant 2



Card 2 — variant 2

---

## card-2 — variant 3



Card 2 — variant 3

---

## card-3 — variant 1

Card-3 block — three-card layout. Accepts exactly three card ids in `cards`. The `variant` attribute controls the presentation style.

**Signature:** `[card-3 cards=<comma-separated card-ids> variant=<int 1|2|3|4|5> width=<int> caption=<string>]`

**Parameters:**

- `cards` (required, comma-separated card-ids) — Exactly three card ids.
- `variant` (optional, int 1|2|3|4|5, default 1) — Visual layout variant. Each variant carries a default width.
- `width` (optional, int, overrides variant default) — Block width as a percentage of the text area.
- `caption` (optional, string) — Caption text displayed below the cards.

**Examples:**



Card 3 — variant 1

---

## card-3 — variant 2





Card 3 — variant 2

---

## card-3 — variant 3



Card 3 — variant 3

---

## card-3 — variant 4





Card 3 — variant 4

---

## card-3 — variant 5



Card 3 — variant 5

---

## card-4 — variant 1

Card-4 block — four-card layout. Accepts exactly four card ids in `cards`. The `variant` attribute controls the presentation style.

**Signature:** `[card-4 cards=<comma-separated card-ids> variant=<int 1|2|3|4|5> width=<int> caption=<string>]`

**Parameters:**

- `cards` (required, comma-separated card-ids) — Exactly four card ids. - `variant` (optional, int 1|2|3|4|5, default 1) — Visual layout variant. Each variant carries a default width. Variants 4 and 5 use a stacked 2x2 layout. - `width` (optional, int, overrides variant default) — Block width as a percentage of the text area. - `caption` (optional, string) — Caption text displayed below the cards.

**Examples:**



Card 4 — variant 1

---

## card-4 — variant 2

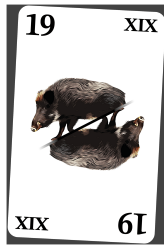


Card 4 — variant 2

---

## card-4 — variant 3





Card 4 — variant 3

---

## card-4 — variant 4 (stacked 2x2)



Card 4 — variant 4 (stacked)

---

## card-4 — variant 5 (stacked 2x2)





Card 4 — variant 5 (stacked)

---

## ## contract-table

Pulls the contract table for a game from the SSOT and renders it as a formatted table. The `game` attribute identifies the game and `scope` selects the rule-set variant.

**Signature:** `[contract-table game=<string> scope=<string>]`

**Parameters:**

- `game` (required, string) — Game identifier (e.g. `zwanzigerrufen`). - `scope` (required, string) — Scope label (e.g. `basic`, `advanced`).

**Examples:**

zwanzigerrufen

Contract	Minimum points	Victory points
Normal Game	45	1
<i>Partnership contract. Forehand calls a secret partner by naming the holder of Tarok XX (or XIX, XVIII etc. if forehand holds the higher cards themselves). Partnership is revealed only through play. Win condition: declaring side collects &gt; 44 card_points. Tie at 44-44 is a loss for the forehand side.</i>		
Solo	45	4
<i>Declarer plays alone against all three opponents. Win condition: declarer collects &gt;= 45 card_points.</i>		
Farberl	45	4
<i>Solo contract with inverted trump rules: Tarok cards do NOT beat colour-suit cards. A player may NOT play a Tarok if they still hold any colour-suit cards. Tarok may only be played when it is the player's sole remaining suit. Because Tarok holds no special power, Pagat Último is impossible in Farberl (excluded from its applies_to). Trull, however, remains valid: it is a hand-composition bonus (holding Sküs + Mond + Pagat) and does not require winning tricks, so the trump inversion does not affect it. Win condition: declarer collects &gt;= 45 card_points. Lead-protection: when a tarok is led, the highest tarok played wins. The trump_override (suits beat trumps) applies only to colour-led tricks, not to tarok-led tricks. Sküs led always wins; Mond led wins unless an opponent holds Sküs; etc.</i>		
Valat (announced)		12
<i>Declared as a contract bid before play. Can be played as solo or as a partnership (using the call-the-XX mechanic, same as Normal Game). Win condition: the declaring side wins every single trick. No special rule changes beyond standard play. Rare — only viable with an exceptional hand.</i>		

---

## ## score-table

Pulls the scoring table for a game from the SSOT and renders it. Use `game` to identify the game; the table covers round and game-level scoring formulas.

**Signature:** `[score-table game=<string>]`

**Parameters:**

- `game` (required, string) — Game identifier.

**Examples:**

zwanzigerrufen — Scoring

Outcome	VP
---------	----

---

## ## bonus-list

Pulls the bonus-condition list for a game and scope from the SSOT and renders it as a formatted list.

**Signature:** `[bonus-list game=<string> scope=<string>]`

**Parameters:**

- `game` (required, string) — Game identifier. - `scope` (required, string) — Scope label.

**Examples:**

---

## ## deck-overview

Pulls the deck composition and card-point summary from the SSOT and renders a structured overview block for the named deck.

**Signature:** `[deck-overview deck=<string>]`

**Parameters:**

- `deck` (required, string) — Deck identifier (e.g. `alpine-tarok`).

**Examples:**

## Alpine Tarok

Total cards: 0

	Suit	Cards
Hearts		
Diamonds		
Spades		
Clubs		
Tarock (Trumps)		

---

## ## glossary

Pulls one or more glossary terms from the SSOT vocabulary and renders them as defined-term entries. Pass a comma-separated list of canonical term ids in `terms`.

**Signature:** `[glossary terms=<string>]`

**Parameters:**

- `terms` (required, string) — Comma-separated canonical term ids (e.g. `trump`, `declarer`).

**Examples:**

---

## ## columns

Renders a multi-column layout. The content between ``[columns]`` and ``[/columns]`` is divided into columns using ``[col]`` as the column separator. The ``count`` attribute sets the number of columns.

**\*\*How to write a columns block:\*\*** Wrap the entire multi-column content in a ``[columns count=N]`` ... ``[/columns]`` pair. ``[columns]`` opens the container and begins the first column. Each ``[col]`` tag starts a new column — it is a divider, not a wrapper. ``[/columns]`` closes the last column and the container. Text flow resumes at full page width after ``[/columns]``. For a two-column layout, you need exactly one ``[col]`` between ``[columns]`` and ``[/columns]``.

**\*\*Signature:\*\*** ``[columns count=<int>]...[col]...[/columns]``

**\*\*Parameters:\*\***

- ``count`` (required, int) — Number of columns (e.g. ``2``, ``3``).

**\*\*Examples:\*\***

Left column content — first passage.

Right column content — second passage.

First column text.

Second column text.

Third column text.

---

## ## scene

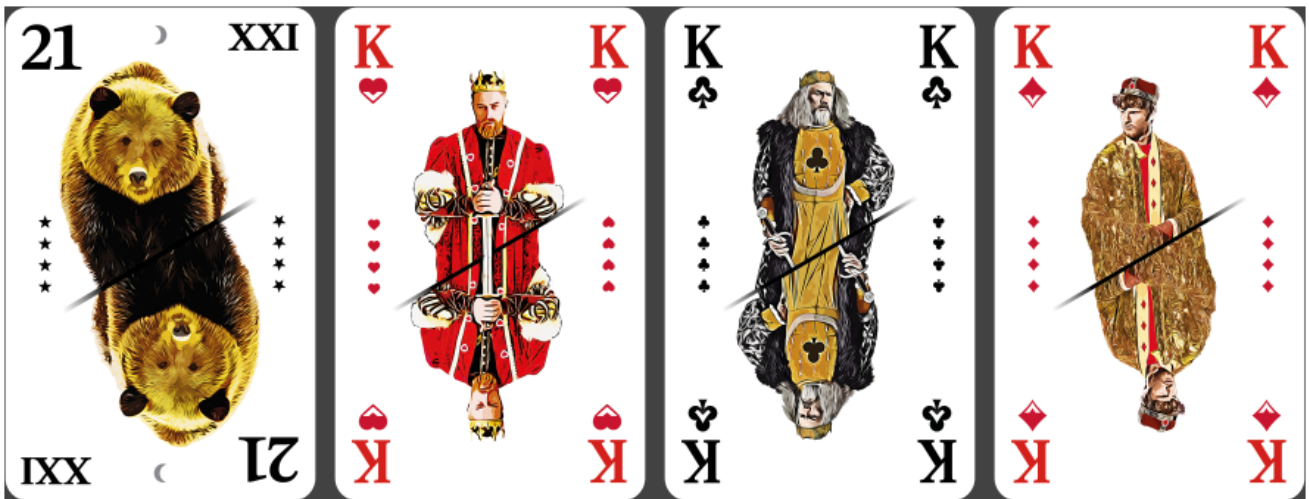
Renders a pre-composed trick or hand scene using the Composer. The ``type`` attribute selects the scene layout; ``cards`` provides the card ids. An optional ``badge`` overlays a small icon and an optional ``caption`` appears below.

**\*\*Signature:\*\*** ``[scene type=<string> cards=<comma-separated card-ids> badge=<string> caption=<string>]``

**\*\*Parameters:\*\***

- ``type`` (required, string) — Scene layout type (e.g. ``trick-scene``, ``hand-scene``, ``talon-scene``). - ``cards`` (required, comma-separated card-ids) — Cards to display in the scene. - ``badge`` (optional, string) — Badge/icon overlay identifier. - ``caption`` (optional, string) — Caption text shown below the scene.

**\*\*Examples:\*\***



Trick scene — four players



---

## pov

Point-of-view callout block for a named player role. Body text is wrapped in a styled box. Use `player` for the role label and `color` for the accent colour token.

**Signature:** `** [pov player=<string> color=<string>]...[/pov]`

**Parameters:**

- `player` (required, string) — Player role label (e.g. `Declarer`, `Defender`).
- `color` (required, string) — Named colour token (e.g. `blue`, `red`).

**Examples:**

As declarer your primary goal is to reach the contract threshold.

As a defender, try to deny the declarer the minimum card-point threshold.

---

## ## sticky

Sticky callout note. Body text is wrapped in a highlighted aside box. No required attributes; the content is supplied in the block body.

**\*\*Signature:\*\*** `[sticky]...[/sticky]`

**\*\*Parameters:\*\***

\_(none)\_

**\*\*Examples:\*\***

Note

**\*\*Tip:\*\*** Always lead with a high trump when you hold more than five.

Note

**\*\*Note:\*\*** The Sküs is always the highest trump regardless of contract type.

---

## ## bid-script

Multi-line dialogue block for auction-based tutorials. Each line uses `Speaker: text` format. Multi-speaker lines like `South / West:` are supported. The shorthand `(others pass)` renders muted/italic. Inline `[CARDID]` tags expand to card images inside the dialogue.

**\*\*Signature:\*\*** `[bid-script]...[/bid-script]`

**\*\*Parameters:\*\***

\_(none — the entire content is the dialogue body)\_

**\*\*Examples:\*\***

**South:** "Rufer."

**West:** "Pass."

**North / East:** "Pass."

**South:** "I call the [t20]."

*(others pass)*

**South:** "Solo."

**West:** "Pass."

**North:** "Pass."

**East:** "Pass."

---

### ## trick-table

Compact multi-trick summary table. Body uses standard markdown table syntax. Cells matching card-id patterns auto-expand to card images. Capture cells matching `+N to (declaring|defending|defenders)` receive colour-token styling.

**Signature:** `[trick-table]...[/trick-table]`

**Parameters:**

\_(none — the content is a markdown table)\_

**Examples:**

#	Lead	South	West	North	East	→
1						+8 to declaring
2						+7 to defending
3						+5 to declaring

---

### ## capture-marker

Per-trick point-capture star rendered inline next to a `[scene type=trick-scene]`. The `points` value shows the card-points taken; `to` indicates which side captured them.

**Signature:** `[capture-marker points=<int> to=<string>]`

**Parameters:**

- `points` (required, int) — Number of card-points captured in this trick. - `to` (required, string) — Capturing side: `declaring` or `defending`.

**Examples:**



to declaring



+7

to defenders

---

## ## stats-row

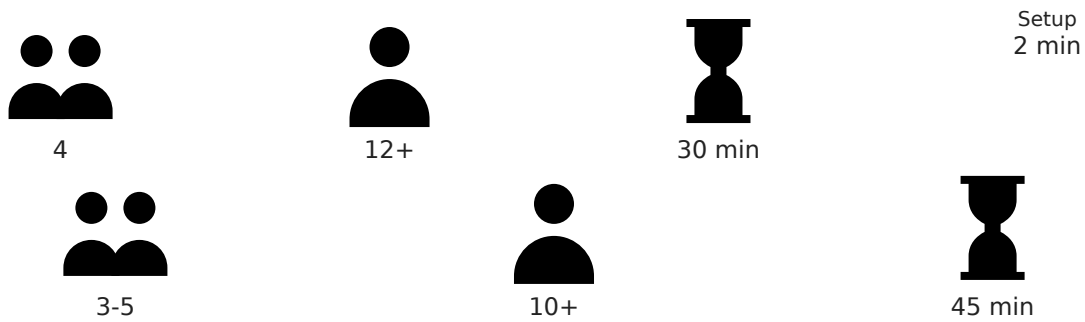
N-cell cover stats row with icons for standard keys (`players`, `ages`, `duration`). Non-standard keys render text-only with humanised labels. All visual tokens are preset-driven. Pass any number of key=value pairs as attributes.

**Signature:** `[stats-row players=<string> ages=<string> duration=<string> ...]`

**Parameters:**

- `players` (optional, string) — Player count or range (e.g. `4`, `3-5`). - `ages` (optional, string) — Recommended age (e.g. `12+`). - `duration` (optional, string) — Typical play time (e.g. `30 min`). - (any key) (optional, string) — Additional stat rendered as text with humanised label.

**Examples:**



---

## ## score-panel

Structural score panel for two named players. Renders an empty scored-round grid. Use in tutorials and explainers where a visual scoring ledger is needed without live data.

**Signature:** `[score-panel p1=<string> p2=<string>]`

**Parameters:**

- `p1` (required, string) — Display name for player 1. - `p2` (required, string) — Display name for player 2.

**Examples:**

---

## ## cta-banner

Full-width call-to-action banner with a link. The `href` attribute sets the destination URL.

**Signature:** `[cta-banner href=<string>]`

**Parameters:**

- `href` (required, string) — Destination URL (e.g. `https://elsteiner.com`).

**\*\*Examples:\*\***

---

**## ref**

Cross-reference anchor and back-reference inline tag. When placed in text it inserts an anchor or a clickable back-reference. Either `section` (intra-document) or `game` (cross-document) must be supplied.

**\*\*Signature:\*\*** ``

**\*\*Parameters:\*\***

- `section` (optional, string) — Intra-document section id to anchor or link. - `game` (optional, string) — Cross-document game id for a game-level reference.

\_(At least one of `section` or `game` is required; enforced by the resolver in Phase 2.)\_

**\*\*Examples:\*\***

This is the anchor: .

Back-reference: .

---

**## inline card shortcode** (`[CARDID]`)

Inline card glyph. Drop `[CARDID]` anywhere in body text or inside another block's body to render a small inline card image at body-text height. Use the canonical card ID per the deck's notation (`[t21]` for Tarokk 21, `[hK]` for Hearts King, `[Skus]` for the Sküs). Useful inside `bid-script` dialogue, `trick-table` cells, and prose explanations of specific cards.

**\*\*Signature:\*\*** `[CARDID]` or `[c id=<card-id> deck=<string>]`

**\*\*Parameters:\*\***

- `id` (required, card-id) — Canonical card identifier. - `deck` (optional, string) — Deck identifier; defaults to the manuscript's default deck.

**\*\*Examples:\*\***

In Königrufen, the Tarokk family runs from [t1] through [t21], with [Skus] as the special trump. Suit cards use single-letter rank: [hK] for Hearts King, [dQ] for Diamonds Queen.

---

**## pagebreak**

Inserts a hard page break. Subsequent content begins on the next page. No attributes.

**\*\*Signature:\*\*** `[pagebreak]`

**\*\*Parameters:\*\***

\_(none)\_

**\*\*Examples:\*\***

---

### ## color-swatch-row


Renders a horizontal colour swatch followed by its label and an optional description. Used in design-reference docs to show preset palette tokens; usable in any manuscript that needs to display a colour value alongside its name.

**Signature:** `[color-swatch-row hex=<string> label=<string> description=<string> size=<float>]`

**Parameters:**

- `hex` (required, string) — Colour value in `#rrggbb` form. - `label` (required, string) — Short name displayed next to the swatch. - `description` (optional, string) — Additional description rendered after the label. - `size` (optional, float) — Swatch size override.

**Examples:**

 **Accent** heading-row backgrounds, accent fills

 **Contract-table border**

---

### ## reset-diagram

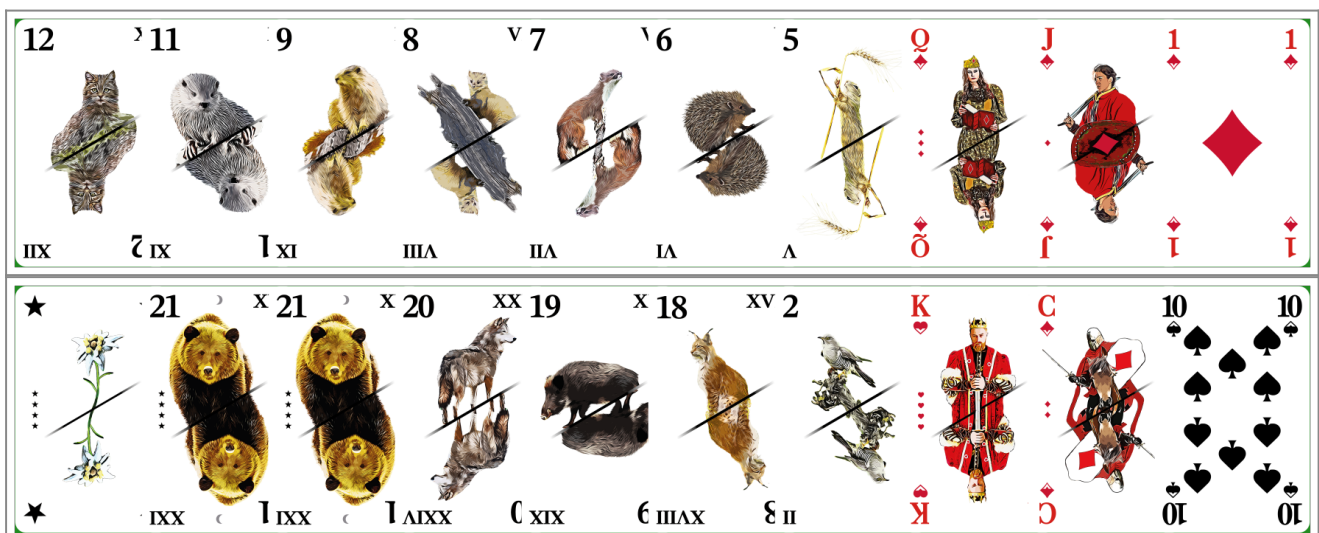
Compass-arranged hand snapshot showing the table position at the start of a new round. `game` identifies the game (must match a `playalong-hands.json` fixture) and `new-dealer` specifies the compass seat of the new dealer as a lowercase string (`north`, `east`, `south`, `west`).

**Signature:** `[reset-diagram game=<string> new-dealer=<string>]`

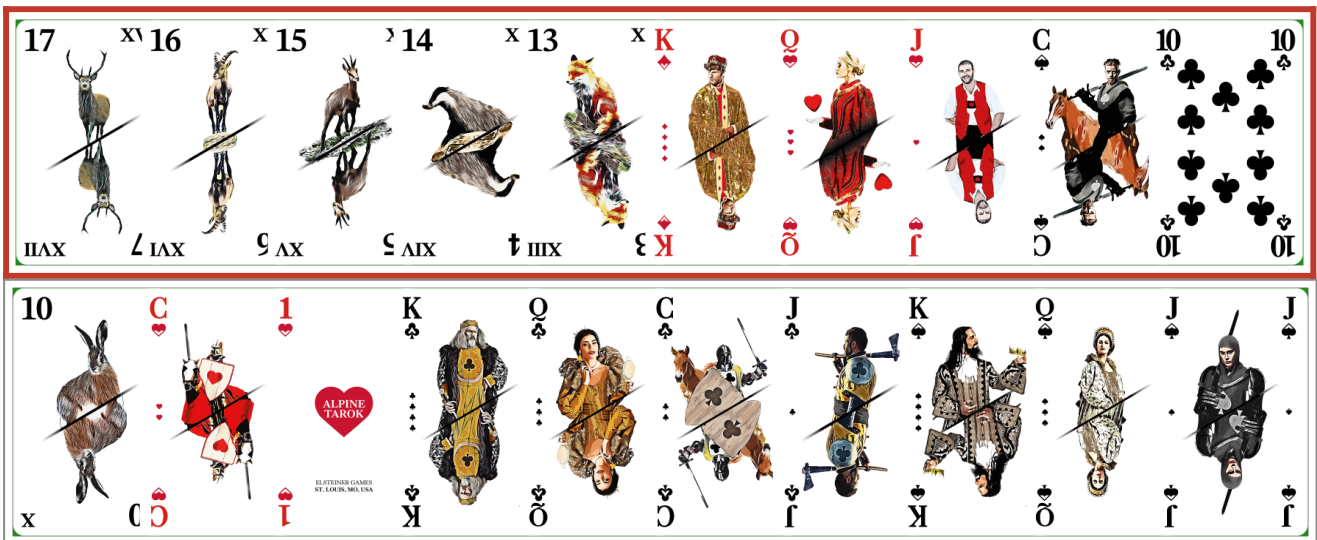
**Parameters:**

- `game` (required, string) — Game identifier matching a snapshot fixture. - `new-dealer` (required, string) — Compass seat of the new dealer (`north`, `east`, `south`, or `west`).

**Examples:**



↻ → East




---

## ## Preset color tokens

The renderer ships with three live presets: `rulebook`, `at-test`, and `playalong`. Each preset's YAML at `renderer/presets/{preset}.yaml` declares per-tag color tokens that drive the visual treatment. Below: the values currently shipped.

### ### rulebook (A4, DejaVu Sans 10pt)

- Accent: `#4a6fa5` (heading-row backgrounds, accent fills)

 **Accent** heading-row backgrounds, accent fills

- Contract-table border: `#222222`

 **Contract-table border**

- Bid-script: speaker `#1d3557`, parenthetical `#777777`, background `#fafafa`


 **Bid-script speaker**

 **Bid-script parenthetical**

 **Bid-script background**

### ### at-test (A4, DejaVu Sans 10pt)

- Accent: `#4a6fa5` (heading-row background)

 **Accent** heading-row background

- Contract-table border: `#222222`

 **Contract-table border**

### ### playalong (A5, DejaVu Sans 9pt)

- Contract-table border: `#222222`



**Contract-table border**

- Bid-script: speaker `#1d3557`, parenthetical `#777777`, background `#fafafa`



**Bid-script speaker**



**Bid-script parenthetical**



**Bid-script background**

- Stats-row: icon `#1d3557`, caption `#1d3557`



**Stats-row icon**



**Stats-row caption**

- Trick-table: header `#1d3557`, declaring `#1d3557`, defending `#a02020`, gridline `#cccccc`, cell `#222222`



**Trick-table header**



**Trick-table declaring**



**Trick-table defending**



**Trick-table gridline**



**Trick-table cell**

- Capture-marker: star `#f4a72b`, points `#222222`, caption `#222222`



**Capture-marker star**



**Capture-marker points**



**Capture-marker caption**

## Glossary

trump (declarer, kitty)