

WENYUAN-CAMPAIGN GENERATED DOCUMENTATION

MAIN MODULE

These are the functions for the main campaign module, which is recommended to be imported into the global scope (i.e. with `*`). For initialisation options, see the `conf` function details.

This documentation was generated using tidy.

- `begin-item()`
- `begin-stat()`
- `bump()`
- `comment-box()`
- `conf()`
- `dndtable()`
- `drop-paragraph()`
- `fancy-comment-box()`
- `make-title()`
- `namedpar()`
- `namedpar-block()`
- `readaloud()`
- `sctitle()`
- `set-theme-colour()`

BEGIN-ITEM

Begins the item environment. See **Item module** documentation.

PARAMETERS

`begin-item`(content)

BEGIN-STAT

Begins the monster statblock environment. See **Statblock module** documentation.

PARAMETERS

`begin-stat`(content)

BUMP

Manually does a 1em paragraph space

PARAMETERS

`bump`()

COMMENT-BOX

A theme-coloured plain comment box

PARAMETERS

```
comment-box(  
  title: content,  
  content: content  
)
```

title `content`

Will be shown in bold small caps

Default: []

CONF

Main configuration function. Use `#show conf.with()`.

PARAMETERS

```
conf(  
  doc,  
  fontsize: length,  
  main-font: array,  
  title-font: array,  
  sans-font: array,  
  sans-smallcaps-font: array,  
  dropcap-font: array  
)
```

fontsize `length`

Main body font size (default 10pt)

Default: default-fontsize

main-font `array`

Body text (default tex gyre and kinghwa)

Default: default-main-fonts

title-font `array`

Title text (default tex gyre and kinghwa)

Default: default-title-fonts

sans-font `array`

Font used in readalouds and comment boxes (default scaly sans)

Default: default-sans-fonts

sans-smallcaps-font `array`

The default sans-font doesn't have smallcaps built in in a way typst knows so scaly sans smallcaps is separate

Default: default-sans-sc-fonts

dropcap-font `array`

For the main chapter drop capitals (default Royal Initialen)

Default: default-dropcap-font

DNDTABLE

Creates a dnd-formatted table.

The use of this table is identical to the default table() interface, EXCEPT you do not have access to stroke, fill, or inset

PARAMETERS

```
dndtable(  
  columns: auto int relative fraction array ,  
  rows: auto int relative fraction array ,  
  gutter: auto int relative fraction array ,  
  column-gutter: auto int relative fraction  
array ,  
  row-gutter: auto int relative fraction array ,  
  align: auto array alignment function ,  
  ..children: content  
)
```

DROP-PARAGRAPH

Makes a paragraph with a drop capital. *N.B.* Since this is rendered in a block unless you have the new typst feature where every paragraph shall be indented you'll need to #bump() the next paragraph.

PARAMETERS

```
drop-paragraph(  
  small-caps: string content ,  
  body: content  
)
```

small-caps `string` or `content`

any text which you wish to be rendered in small caps, like how DnD Does it

Default: ""

FANCY-COMMENT-BOX

A theme-coloured fancy comment box with decorations

PARAMETERS

```
fancy-comment-box(  
  title: content ,  
  content: content  
)
```

title `content`

Will be shown in bold small caps

Default: []

MAKE-TITLE

Makes a simple title page

Parameters:

- title: main book title
- subtitle: (optional) subtitle
- author: (optional)
- date: (optional) – just acts as a separate line, can be used for anything else
- anything-before: (optional) this is put before the title
- anything-after: (optional) this is put after the date
- page-background: (optional)
- text-colour: (optional) the colour the title and bars will be rendered in

PARAMETERS

```
make-title(  
  title: content ,  
  subtitle: content ,  
  author: content ,  
  date: content ,  
  anything-before: content ,  
  anything-after: content ,  
  page-background: content ,  
  text-colour: color  
)
```

subtitle `content`

Appears in smaller font within the bars under the title

Default: []

author `content`

Appears beneath the main title

Default: []

date `content`

Appears beneath the author

Default: []

anything-before `content`

Appears before the main title

Default: []

anything-after `content`

Appears after everything else

Default: []

text-colour `color`

The primary title colour

Default: colours.dndred

NAMEDPAR

A paragraph with a bold italic name at the start

PARAMETERS

```
namedpar(  
  title: str content,  
  content: content  
)
```

title str or content

the bold italic name, a full stop / period is put immediately after for you

NAMEDPAR-BLOCK

See namedpar but this one is in a block environment

PARAMETERS

```
namedpar-block(  
  title: str content,  
  content  
)
```

content
content

READALOUD

A tan coloured read-aloud box with some decorations

PARAMETERS

```
readaloud(content: content)
```

SCTITLE

makes a small caps title block (e.g. for table titles)

PARAMETERS

```
sctitle(content: content)
```

SET-THEME-COLOUR

sets a theme colours from the colours package of this module or any other colour you want, on you if it looks bad :) The colours recommended are: phbgreen, phbcyan, phbmauve, phbtan, dmglavender, dmgoral, dmgsategrey (-ay), dmglilac

PARAMETERS

```
set-theme-colour(colour: color)
```

STATBLOCK MODULE

All functions required for statblocks.

Important. By default these are imported under the subpackage stat. If you import all functions from wenyuan, then you can just call stat.function immediately to access.

Section headers such as *Actions* or *Reactions* are done using the second-level header ==

Action names — the names that go in front of actions / abilities are done using the third level header === (do not leave a blank line between the header and its body text)

- *ability()*
- *challenge()*
- *dice()*
- *dice-roll()*
- *mainstats()*
- *skill()*
- *statheading()*
- *stroke()*

ABILITY

calculates and properly displays ability scores

PARAMETERS

```
ability(  
  str: int,  
  dex: int,  
  con: int,  
  int: int,  
  wis: int,  
  cha: int  
)
```

CHALLENGE

takes an integer CR and formats with experience will do nothing if CR is not a standard number represent non integer CRs as decimals

PARAMETERS

```
challenge(cr: int)
```

DICE

Calculates and displays a DnD dice average from a string formatted roll accepts strings of the following form:

- 0d0 (where 0 means any integer number)
- 0d0+0
- \d+d\d+(\+\d+)?

and will accept strings with any number of spaces (but no other characters) it is the user's responsibility to ensure that the formatting is correct

PARAMETERS

```
dice(value: str)
```

DICE-RAW

displays a DnD dice average format, e.g. 19 (3d6 + 9) given the number of dice, the sides of dice, and a modifier (which can be set to 0 for no modifier)

Not necessarily recommended to be used directly; use dice for an easier interface.

PARAMETERS

```
dice-raw(  
  num-dice: int,  
  dice-face: int,  
  modifier: int  
)
```

MAINSTATS

AC, HP, Speed stats as one generated block.

Expects hp_dice as a valid dice value. If you don't want to use this just use hp_etc

PARAMETERS

```
mainstats(  
  ac: str,  
  hp-dice: str,  
  speed: str,  
  hp-etc: str  
)
```

ac str
Armour class
Default: ""

hp-dice str
Dice amount for HP, which will be calculated
Default: ""

speed str
Speed
Default: "30ft"

hp-etc str
Freeform HP text, can be used in conjunction or not
Default: ""

SKILL

a skills or stats entry like:

- **Hit Points** 60 (8d8 + 24)
- **Senses** Passive perception 15

the title is the thing in bold and the contents can be anything

PARAMETERS

```
skill(  
  title: content,  
  contents: content  
)
```

STATHEADING

Header block for monster stats.

PARAMETERS

```
statheading(  
  title-text: str content,  
  desc: content  
)
```

title-text str or content
Usually for the name of the monster.

desc content
The little italic bit of description that says stuff like *Medium undead, lawful evil*.
Default: []

STROKE

Draws a stroke

PARAMETERS

```
stroke()
```

ITEMS MODULE

All functions required for basic items.

Important. By default these are imported under the subpackage `item`. If you import all functions from `wenyuan`, then you can just call `item.function` immediately to access.

Item Name is done with the top-level header =

Section headers are the second level header ==

Abilities and named paragraphs are the third level header ===

- `flavourtext()`
- `smalltext()`

FLAVOURTEXT

An italic block quote for the flavour text of this item

PARAMETERS

`flavourtext`(content)

SMALLTEXT

If you need any small text

PARAMETERS

`smalltext`(content)

COLOURS MODULE

A preset list of default colours you can use.

Important. By default these are imported under the subpackage `colour`.

It's all variables so I'll just copy it here:

// based on <https://github.com/rpgtex/DND-5e-LaTeX-Template/blob/dev/lib/dndcolors.sty>

// page

`#let bgtan = rgb("#F7F2E5")` // readAloud

`#let pagegold = rgb("B89A67")` // numbering

`#let clear = rgb("ffffff00")`

// type

`#let dndred = rgb("#58180d")`

`#let rulegold = rgb("#C9AD6A")` // subsection rule

`#let shadow = rgb("AAAAAA")`

// trim

`#let brgreen = rgb("#E8E6DC")` // Basic Rules

`#let phbgreen = rgb("#E0E5C1")` // PHB Part 1

`#let phbcyan = rgb("#B5CEB8")` // PHB Part 2

`#let phbmauve = rgb("#DCCCC5")` // PHB Part 3

`#let phbtan = rgb("#E5D5AC")` // PHB appendix

`#let dmglavender = rgb("#E3CED3")` // DMG Part 1

`#let dmgoral = rgb("#F3D7C1")` // DMG Part 2

`#let dmgsategrey = rgb("#DBE4E4")` // DMG Part 3

`#let dmglilac = rgb("#D7D4D6")` // DMG appendix

`#let dmgsategray = dmgsategrey`