

# The `dndicons` package

A set of high quality icons made with TikZ for use in notes for tabletop role-playing games

Jasper Habicht<sup>1</sup>

Version 1.3.2, released on 19 February 2024

---

## 1 Introduction

The `dndicons` package provides a set of high quality icons made with TikZ for use in notes for tabletop role-playing games. The icons are meant to be used in the body text, but they can also be used in other contexts such as graphics or diagrams.

Since the commands to typeset the icons typeset `tikzpicture` environments, these commands should not be used inside another `tikzpicture`. However, because the package defines the icons as TikZ shapes, it is possible to use the icons in `tikzpicture` environments directly. Apart from that, as of version 1.1.0, the package provides a way to define custom commands to typeset the icons as boxed material which is safe to use in a `tikzpicture` context.

## 2 Loading the package

The `dndicons` package is loaded by calling `\usepackage{dndicons}` in the preamble of the document.

```
pics
```

The package provides the option `pics`. If the package is loaded with this option, every icon is also available as TikZ pic. On the use of `pics`, see section 3.4 below.

### 2.1 Dependencies

The package loads the `tikz` package which in turn loads the `xcolor` package. If you want to make use of certain options these packages provide, you need to load the packages with the relevant options beforehand or use, for example, `\PassOptionsToPackage{svgnames}{xcolor}`.

## 3 Usage

Once loaded, the package provides a set of commands that can be used to print icons inline. The package also defines a set of node shapes that can be used inside a `tikzpicture` environment.

---

<sup>1</sup> E-mail: [mail@jasperhabicht.de](mailto:mail@jasperhabicht.de)

### 3.1 Global settings

`dnd icons`

All icons share the TikZ style `dnd icons` that has no options per default but can be used to style all icons at once. For example, if the setting `\tikzset{dnd icons/.append style={draw=red}}` is placed at the beginning of the document, all icons will be drawn in red. Per default, the icons are drawn in the color of the surrounding text.

Note that it may be necessary to add the TikZ option `transform shape` when applying transformations to the icons, because the icons are realised as TikZ nodes which are not affected by some transformations per default.

`dnd icons/background color`

Some icons can be used with a negative color scheme where the icon is drawn negatively inside a filled shape. Per default, the icons are drawn in white in such cases, but it might be desirable that the icons are in the same color as the background. To this end, the color can be changed using the TikZ option `dnd icons/background color` in the following way:





```
\colorbox{blue!50}{%
  \ability[negative]{charisma}
  [scale=2, transform shape]%
}

\tikzset{
  dnd icons/background color={blue!50}
}
\colorbox{blue!50}{%
  \ability[negative]{charisma}
  [scale=2, transform shape]%
}
```

This feature can, of course, also be used to change the color of the icon independently from the color of the background.

`dnd icons/before sep`  
`dnd icons/after sep`

The TikZ options `dnd icons/before sep` and `dnd icons/after sep` are used to define the width of the space that is added before and after the icons respectively. The default value of both lengths is 0.05 em. For example, setting the space before icons to 1 cm can be achieved as follows:

Roll  a die!  
Roll  a die!

```
Roll\die{eightside}{}a die!

\tikzset{
  dnd icons/before sep={1cm}
}
Roll\die{eightside}{}a die!
```

### 3.2 Icons

Because the package defines the icons as TikZ shapes, every command can actually be used together with every shape. However, the combinations of shapes and commands as described in the follow-


ing subsections are preferable.


### 3.2.1 Icon `\die`


`\die`[*<style>*]{*<shape>*}[*<options>*]{*<integer>*}





















The command `\die` prints an icon to depict a die with a certain number of sides. Two special icons exist for a two-sided die (which would be equivalent to a coin) and for a hundred-sided die (which typically comes in the shape of a sphere). There is also a special icon for a fudge die.

For the six-sided die, nine additional shapes exists representing the values one to nine as pips. Also, additional shapes exist representing the plus or minus side of a fudge die.

The command takes two mandatory commands, the first of which describes the shape (see previous subsection) and the second can take an integer that is placed in front of the shape. Thus, `\die{eightside}{2}` results in 2 (meaning 2 eight-sided dice are rolled).

The command also takes two optional arguments, the second of which can take arbitrary TikZ options to style the icon. The options affect the shape, not the integer when it is printed before the icon. As an example, `\die{eightside}[blue, thick]{2}` results in 2.


The first optional argument can take the value `normal` or `large`, `normal` being the default value. With `large` given as argument, the icon is drawn larger and the additional integer is printed inside of the shape instead of before it. As an example, `\die[large]{eightside}{2}` results in . Note that the integer will always be printed on top of the shape, even if the shape does not have an open center as in the case of the `fudge` shapes or the shapes featuring pips.

Command	Icon	Shape
<code>\die</code>		<code>twoside</code>
		<code>fourside</code>
		<code>sixside</code>
		<code>eightside</code>
		<code>tenside</code>
		<code>twelveside</code>
		<code>twentyside</code>
		<code>hundredsideside</code>
		<code>fudge</code>
		<code>sixside one</code>
		<code>sixside two</code>
		<code>sixside three</code>
		<code>sixside four</code>
		<code>sixside five</code>
		<code>sixside six</code>
		<code>sixside seven</code>
		<code>sixside eight</code>
		<code>sixside nine</code>
		<code>fudge plus</code>
		<code>fudge minus</code>

### 3.2.2 Icons `\ability` and `\saving`


`\ability`[*<style>*]{*<shape>*}[*<options>*]




















The command `\ability` prints icons depicting an ability of a character. The abilities are represented by animal-like shapes. The relevant shape should be given as mandatory argument to the command. The second optional argument can take arbitrary TikZ options to style the icon.

The first optional argument can take the value `positive` or `negative`, `positive` being the default value. With `negative` given as argument, the icon is drawn negative inside a circle. As an example, `\ability[negative]{charisma}` results in .

`\saving[<style>]{<shape>}[<options>]`

The command `\saving` prints the icons that can be typeset using the `\ability` command inside a small shield. It can take the same values for the mandatory argument as the `\ability` command. The optional argument can take arbitrary TikZ options to style the icon.










The first optional argument can take the value `normal` or `empty`, `normal` being the default value. With `empty` given as argument, the icon inside the shield is not printed. In this case, the mandatory argument can be left empty. As an example, `\saving[empty]{}`  results in .

Command	Icon	Shape
\ability		strength
		dexterity
		dexterity alt
		constitution
		intelligence
		wisdom
		charisma
		luck
		armor
\saving		strength
		dexterity
		dexterity alt
		constitution
		intelligence
		wisdom
		charisma
		luck
		armor
\saving		proficiency

### 3.2.3 Icon `\spell`

`\spell{<shape>}[<options>]`

The command `\spell` prints icons depicting the effect of a spell or how it is to be effected. The optional argument can take arbitrary TikZ options to style the icon.


Command	Icon	Shape
\spell		linear
		conic
		quadratic
		cubic
		spheric
		cylindric
		verbal
		somatic
		material









Command	Icon	Shape
		focus

### 3.2.4 Icon `\spellschool`

`\spellschool`[*<style>*]{*<shape>*}[*<options>*]

The command `\spellschool` prints icons that represent the school a spell belongs to. The second optional argument can take arbitrary TikZ options to style the icon.

The first optional argument can take the value `negative` or `positive`, `negative` being the default value. Per default the icon is drawn in white inside a filled escutcheon. With `positive` given as argument, the icon as well as the escutcheon are drawn in the currently selected color. As an example, `\spellschool[positive]{evocation}` results in .

Command	Icon	Shape
<code>\spellschool</code>		abjuration
		conjuration
		divination
		enchantment
		evocation
		illusion
		necromancy
		transmutation

### 3.2.5 Icons `\damage`, `\attack`, and `\condition`

`\damage`{*<shape>*}[*<options>*]









The command `\damage` prints icons depicting the damage of an attack. The icon is printed inside a circle. The optional argument can take arbitrary TikZ options to style the icon.

`\attack`{*<shape>*}[*<options>*]

The command `\attack` prints icons depicting the kind of an attack. The optional argument can take arbitrary TikZ options to style the icon.

`\condition`{*<shape>*}[*<options>*]

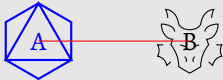
The command `\condition` prints icons depicting a condition of a character. The optional argument can take arbitrary TikZ options to style the icon.

Command	Icon	Shape
<code>\damage</code>		acid
		bludgeoning
		cold
		fire
		force
		lightning
		necrotic
		piercing

Command	Icon	Shape
		poison
		psychic
		radiant
		slashing
		thunder
		healing
<code>\attack</code>		melee
		ranged
		magic
		singlehanded
		doublehanded
<code>\condition</code>		buff
		blinded
		charmed
		deafened
		exhausted
		frightened
		grappled
		incapacitated
		invisible
		paralyzed
		petrified
		poisoned
		prone
		restrained
		stunned
		unconscious
		hearing
		seeing

### 3.3 Direct use of shapes

Because the icons are defined as *TikZ* shapes, they can directly be applied to *TikZ* nodes. However, the shapes don't have a shape border and no anchors except for the `center` anchor that sits exactly in the center of the shape. Therefore, if nodes with these shapes are connected using edges, the `center` anchor will be used to connect the nodes. If nodes with these shapes are being positioned, only the `center` anchor is available. Text content of these nodes is simply printed on top of the center of the node. Compare the following example.



```

\begin{tikzpicture}
  \node[eightside, blue, thick]
    at (0,0) (A) {A};
  \node[charisma] at (2,0) (B) {B};
  \draw[red] (A) -- (B);
\end{tikzpicture>

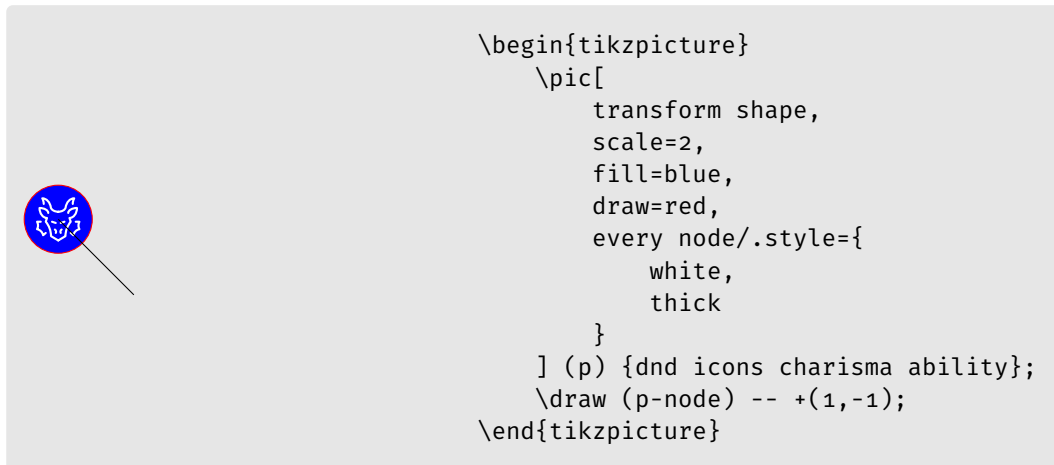
```

### 3.4 Icons as pics

If the package is loaded with the option `pics`, every icon is also available as *TikZ* pic. The names of the pic always start with `dnd icons` followed by a space and the name of the relevant icon (see the

lists above). For abilities, savings, spellschools and damages, additional pics exists where the name has the suffixes `ability`, `saving`, `spellschool`, and `damage` respectively.

The icon is embedded as a node in the pic which has the name `-node`. Thus, it is possible to name the pic and refer to the node inside. Due to the fact that the icon is a node, the option ‘transform shape’ has to be used if transformations on the pic are to affect the node as well. It is easily possible to apply styles to the node using the TikZ option `every node` as shown in the following example.



```

dnd icons/create pic from shape
dnd icons/create pic from ability shape
dnd icons/create pic from saving shape
dnd icons/create pic from spellschool shape
dnd icons/create pic from damage shape

```

The package defines five TikZ keys that are used to create pics using the relevant node shapes. In normal circumstances, it is not necessary to use these keys. They are mentioned here primarily for reference.

### 3.5 Boxing of icons


Because the icons cannot simply be used inside `tikzpicture` environments, the package provides a workaround to place icons inside of boxes for later use. Icons that are boxed this way can safely be used inside `tikzpicture` environments. This might be necessary, if an icon should be used in inline text that sits inside a node.

```
\provideprotecteddndicon{<command>}[<style>]{<shape>}[<options>]{<box name>}
```

The command `\provideprotecteddndicon` creates a box containing the icon that would be created using one of the regular commands this package provides.

Using `\provideprotecteddndicon{die}[large]{eightside}[blue, thick]{mybox}`, for example, the icon of an eight-sided die with the relevant style and TikZ options would be stored in a new box named `mybox`. Note that no integer can be added to the `\die` command in this context.

```
\useprotecteddndicon{<box name>}
```

Using the command `\useprotecteddndicon`, the previously defined box can be used to place the relevant icon. With the above definition, `\useprotecteddndicon{mybox}` would result in .

Having created a boxed icon, it is safe to use it, for example, inside a TikZ node:



```
\begin{tikzpicture}
  \node[circle, draw, align=center] {
    \useprotecteddndicon{mybox} \\\
    Roll a die!
  };
\end{tikzpicture}
```

## 4 Changes

### **v1.1.0** (2023/08/15)

First public release.

### **v1.1.1** (2023/11/15)

Fudge dice icon added.

### **v1.1.2** (2023/11/16)

Bug fixed that caused wrong spacing when using dice icons without quantifier.

### **v1.2.0** (2023/11/20)

Corrections in the manual. Icons for six-sided dice with one to nine pips, plus sign and minus sign added.

### **v1.2.1** (2023/11/20)

Documentation of lengths for spacing added.

### **v1.2.2** (2023/11/21)

Corrections in the manual.

### **v1.3.0** (2023/11/21)

Option to set background color added. Renamed global option.

### **v1.3.1** (2024/02/18)

Correction of initializing code. Correction of default value of after sep. Addition of pics.