Syntax

There have been several versions of bofa over time before settling on this one. Currently the syntax is actually just pretty similar to json, but shorter and imo nicer.

```
key value
```

Everything is made up of key value pairs. Keys dont need to be quoted and can contain any character that isnt tab. space, or a line break. However they should generally follow the symbol rules of programming languages (dont start with a number, contain only letters and _)

```
key ( key value another_key value )
```

Objects are indicated using normal brackets. An object is a value type and can be used in place of a value. I decided to use brackets instead of curly brackets, because they are nicer to type of german keyboards.

```
key [ value value valuee ]
```

Arrays use angled brackets. Can also be used in place of values and are not allowed to have keys.

```
#This is a comment
```

Single line comments continue until the next line break.

```
/* Multi line comments */
```

Multi line comments will continue until the ending character sequence '*/' without needing a whitespace inbetween, so you could theoretically start a key or value directly after a ml comment

Comments cannot start at every position. They need to follow a whitespace, so that they are visibly not part of a key. This also means that comment sequences can be part of a key if there is no whitespace character inbetween. This is bad style though and should not actually be done.

```
key 344.23432
```

A number is a type of value. Currently they have only one notation option, however in the future other notation, such as binary should probably be supported. Numbers can contain any amount of 0-9 and up to a single '.' as a seperator. They are not required to contain the '.' to make writing integers easier. If a number includes a '.' it also needs atleast one character infront of it and one character after it. '.5' is not a valid number. Any amount of prepending 0's is allowed.

```
key true
```

A boolean can have two values, true or false. These need to be literal lowercase values. These literal values are not reserved, so keys can technically use them. However that would be bad style and should be avoided.

```
key "This is a string"
```

Double quotes are used for unescapeable strings. Once you start a string with '""' you simply keep going until you hit the next instance. Regardless of any newlines or other characters in the way. This type mostly exists for parsing reasons. If you dont need to transform a string you can parse it quicker.

Revision #1
Created 2026-05-21 18:39:19 UTC by Booklordofthedings
Updated 2026-05-21 21:23:16 UTC by Booklordofthedings