

1 Fun stuff

When I decided to add the `\uuid` macro (a trivial task because there was already a LUA function for it in `CONTEX`) I wondered about other functions that could be added, like those for sine and cosine. There is no real need for that because we can already do this:

```
\luaexpr{math.sin(math.pi/4)}
```

which gives us 0.70710678118655 as result, but still one can ponder the usability of additional macros. When seeing this, one of the first things that probably comes to mind is how to get less digits, and indeed that can be achieved, as 0.7071 demonstrates.

```
\luaexpr[.4N]{math.sin(math.pi/4)}
```

The optional argument between square brackets is a template as we know from other `CONTEX` commands without the leading percentage sign. But what if we don't want this expression and explicit math function call?

```
$ \sin (x) = \the\sin{pi/8} $
```

This gives us a normal rendered sin function symbol at the left hand and a numeric result at the right hand: $\sin(x) = 0.382683$. The nice thing about it is that we don't need to come up with new macro names.¹ In a similar fashion we can do this:

```
$ \sind(x) = \luaexpr[.4N]{math.sind(120)} = \the\sind[.4N]{120} $
```

Both calls give the same result: $\sind(x) = 0.866 = 0.866$ and in case you wonder why we have only three digits: the `N` formatter removes trailing zeros. However, the `\the` prefix is still not that nice, apart from the fact that we abuse a feature of the LUA interface meant for other purposes (read: we cheat). So, in a next step in exploring this I cooked up:

```
$ \sqrt(x) = \the\sqrt[.3N] {2} $  
$ \sqrt(x) = \compute\sqrt[.3N] {2} $
```

There is still a prefix but `\compute` looks more natural. It is not an alias for `\the` but a shortcut for a prefix feature that can drive all kind of interpretations of in this case `\sin`, and that is probably where the real fun will start. Instead of functions we can also have constants:

In case you wonder how extensible this mechanism is, here is what happens in the `mathfun` module that needs to be loaded in the usual way. There you find:

The module also provides a few more expression variants (these can end up in the core if really needed much):

```
$ \pi = \mathexpr[.40N]{pi} $
```

¹ At some point `CONTEX` might introduce a namespace mechanism to deal with possible conflicts between environments.

```

$ \pi = \mathexpr[.80N]{sqrt(11)}      $
$ \pi = \decimalexpr[.80N]{sqrt(11)}  $
$ \pi = \decimalexpr{sqrt(11)}        $
$ c = \complexexpr{123 + new(456,789)} $

```

This gives:

```

π = 3.1415926535897931
π = 3.3166247903553998
π = 3.3166247903553998
π = 3.3166247903553998491149327366706866839270885455894
c = 579 + 789i

```

The question is: do we need this and if so, what more do we need? Feel free to bring it up on the `CONTEX`T mailing list. It anyway is a nice demonstration of what can be done with the mix of languages.