

Unicode test

July 13, 2019

Contents

Contents	i
I Unicode	1

Part I

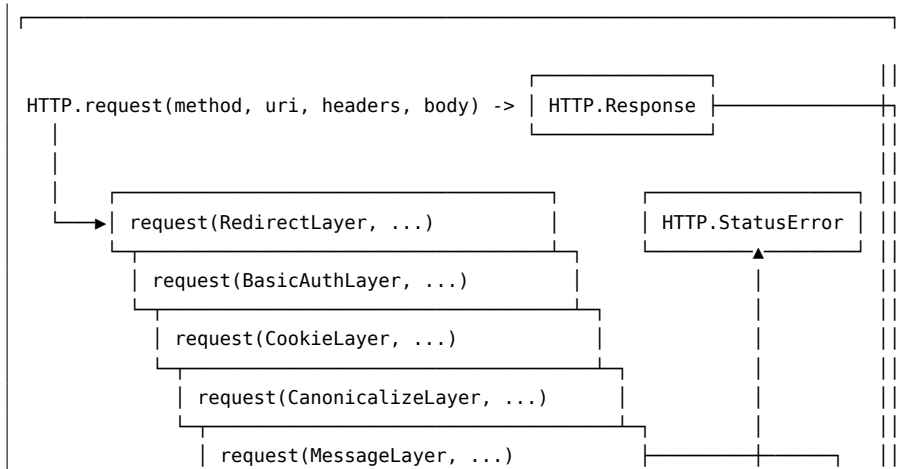
Unicode

Some unicode tests here.

Apart from the ASCII codes, the operator is encoded at U+22BB √ XOR (HTML ⊻) and U+2295 ⊕ CIRCLED PLUS (HTML ⊕ · ⊕), both in block mathematical operators.

In Lato:

- Checkmark: "✓"
- Cycled plus: "⊕"
- XOR: "√"
- Exists: "∃", forall: "∀"



`\begin{lstlisting}` is used for non-highlighted blocks:

```
xor:    √
forall: ∀
exists: ∃
check:  ✓
oplus:  ⊕
```

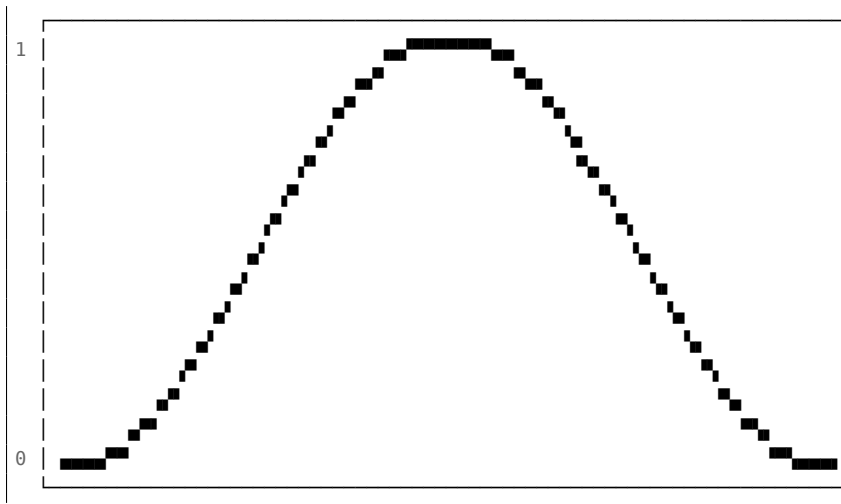
What about the other edge cases: `\ % \ % \ % \ % \ % \ %`

`\begin{minted}` is used for highlighted blocks:

```
xor:    √
forall: ∀
exists: ∃
check:  ✓
```

Inlines:

```
xor: √ -> %\unicodeveebar% <-
forall: ∀, exists: ∃, check: ✓
%\%\unicodeveebar, oplus: ⊕
```



```
2x4 DataFrames.DataFrame|
Row | a      | b      | c      | d      ||
    | Int64 | Float64 | Int64 | String ||-----|
1   | 2     | 2.0   | 2     | John   ||
2   | 2     | 2.0   | 2     | Sally  |
```

```
function map_filter_iterators(xs, init)
  ret = iterate(xs)
  ret === nothing && return
  acc = init
  @goto filter
  local state, x
  while true
    while true
      ret = iterate(xs, state)           # input
      ret === nothing && return acc      #
      @label filter                     #
      x, state = ret                    #
      iseven(x) && break                 # filter :
    end                                 # :
    y = 2x                               # imap :
    acc += y                             # + :
  end                                    # : :
  #                                     + <-- imap <----- filter <-- input
  return acc
```