



Inhaltsverzeichnis

--



```
        end
        if Unicode_data.is_whitespace(codepoint) then
            printed_codepoint = style(printed_codepoint, "background_
        end
        return printed_codepoint
    else
        return ""
    end
end

local function u_plus(codepoint)
    return ("U+%04X"):format(codepoint)
end

local function get_codepoint(args, arg)
    local val = args[arg]
    local is_negative = false

    if type(val) ~= "string" then
        errorf("code point in [[hexadecimal]] expected, got %s", type(val))
    elseif val:find('^%s*%-') then
        -- Negative number strings yield a bizarre value:
        -- tonumber("-1", 16) -> 1.844674407371e+19.
        -- Strip initial minus.
        val = val:match("%-(.+)")
        is_negative = true
    end

    local codepoint = tonumber(val, 16)
        or errorf("code point in [[hexadecimal]] expected, got %q", val)

    if is_negative then
        codepoint = -codepoint
    end

    if not (0 <= codepoint and codepoint <= 0x10FFFF) then
        errorf("code point %d out of range", codepoint)
    end

    return codepoint
end

function p.unichar(frame)
    local args = frame.args[1] and frame.args or frame:getParent().args
    local codepoint = get_codepoint(args, 1)

    local name_or_label = Unicode_data.lookup_name(codepoint)
    local is_label = name_or_label:sub(1, 1) == "<"

    return ("%s %s %s"):format(
        style(u_plus(codepoint), "monospace"),
        show(codepoint),
        is_label and name_or_label or style(name_or_label, "smallcaps"))
end

return p
```