

# {gender\*render}

## Specification

Template system and implementation  
specification for rendering gender-neutral email  
templates with pronoun information

phseiff  
phseiff.com

November 11, 2020

## Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Concept</b>	<b>2</b>

## 1 Abstract

Our society, as well as the way we perceive gender, are steadily evolving. This evolution does not hold in front of technological question, and it is our -the "IT people"-s duty- to address do our best to solve social that arise from our technology. One such technology are email- and other text templates, which are becoming increasingly popular to automate customer interactions of any kind, be it in newsletters, notifications or program menus. Many such templates are gender-specific, meaning that they address the reader in a gender-specific fashion ("Dear Mrs. Dursley, ..."). Those templates relatively easily implemented by providing two versions of the email, one for every binary gender. However, some texts are far more complicated, because they address multiple people (each with their own unknown at the time of writing), or people in the third person

(throwing their pronouns into the mix). In addition, an increasingly height amount of people use non-binary pronouns, or gender-neutral pronouns, many of whom might now yet be discovered at the time of writing, which makes these people marginalized when it comes to being correctly addressed even in automated emails.

This creates the requirement for creating template systems for the english language, and, in extention, any language (since all languages work differently), that support writing complex texts in a gender-neutral fashion and later "render" them to correctly gendered texts.

$\{\text{gender}^*\text{render}\}$  is an attempt at creating one such template language, including a Specification, to serve as a proof of concept as well as a starting point for people who want to implement similar things. The vision behind this proof of concept is not only to show *how* addressing people with unconventional preferred pronouns can be automatized, but also to show *that* it can be easily automatized, to debunk the myth that properly addressing nonbinary people in an automated fashion is simply technically impossible.

## 2 Technical Requirements

There are multiple requirements for such a template language, whom I will list here, including short explanation of why they are required wherever I deem it necessary:

- The language must be easy to use even for less tech affine people. This means that the atoms of the language, such as tags et cetera, must be as short as possible, and should not clash with commonly used words or signs, so the amount of escape characters the user needs to use is minimal.
- The language must support different scenarios:
  - One person being addressed versus multiple people being addressed
  - Only people mentioned in first person, only people mentioned in third person, or a mixture of both
  - Everyone using pronouns versus some people preferring not to use any pronouns
- The fact that multiple scenarios are supported may not make using the template language for only a subset of them more complicated that it needs to be.
- Rendering templates may only require the information needed for rendering the template. For example, rendering a template that never addresses anyone in the first person should not require providing information as to whether the person goes by "Mr", "Mrs" or any other form of address.



---

This is especially relevant since users do not want and should not need to require more information that necessary for rendering the templates, especially considering the intimate nature of preferred pronouns.

- The syntax should be describable using a context-free grammar in conjunctive normal form, which allows easy syntax checking and syntax highlighting.
- The data containing a persons preferred pronouns should be given in a widely-used, standardized format, such as JSON.

### 3 Design Decisions

The following decisions where made based on the the technical requirements ruled out in the corresponding section:

- The language uses a syntax similar to pythons build-in string formatting syntax, using curly brackets to annotate gender-specific parts of a sentence. Backslashes are used as escape characters for the rare occurrences where curly brackets are actually needed.
- In addition to terms like "possessive pronoun", using the gender-neutral form ("their") in tags is supported, potentially making texts more fluid to write and easier to read in their un-rendered form.
- If tags contain IDs to annotate which person is referred to, a mapping of IDs to pronoun preferences is accepted for rendering. If no such IDs are added to the document because only one person of unknown gender is addressed in the document, the pronoun differences are directly accepted by the renderer. This supports referring to multiple persons in one text without making the writing of texts that refer to only one person any more troublesome.
- The pronoun information is given to the renderer as a piece of JSON data (or a dictionary if the language used by the implementation supports it). Information that is not required by the template may be left out in the template.
- Templates can be parsed before being rendered and then used for multiple renderings. This should debunk the idea that such template systems are to inefficient to use them.