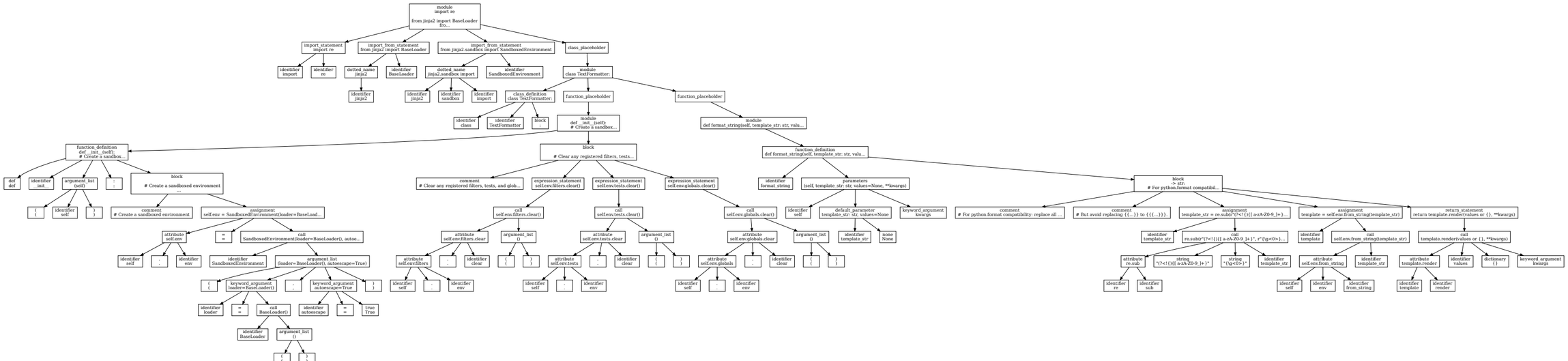
module
import re
from jinja2 import BaseLoader
fro...
import_statement
import re
identifier
import
identifier
re
import_from_statement
from jinja2 import BaseLoader
dotted_name
jinja2
identifier
BaseLoader
identifier
jinja2
import_from_statement
from jinja2.sandbox import SandboxedEnvironment
dotted_name
jinja2.sandbox import
identifier
SandboxedEnvironment
identifier
jinja2
identifier
sandbox
identifier
import
class_placeholder
module
class TextFormatter:
class_definition
class TextFormatter:
identifier
class
identifier
TextFormatter
block
:
function_placeholder
module
def __init__(self):
# Create a sandbox...
function_placeholder
module
def format_string(self, template_str: str, valu...
function_definition
def __init__(self):
# Create a sandbox...
def
def
identifier
__init__
argument_list
(self)
(
(
identifier
self
)
)
:
block
Create a sandboxed environment
...
comment
Create a sandboxed environment
assignment
self.env = SandboxedEnvironment(loader=BaseLoad...
attribute
self.env
identifier
self
.
.
identifier
env
=
=
call
SandboxedEnvironment(loader=BaseLoader(), autoe...
identifier
SandboxedEnvironment
argument_list
(loader=BaseLoader(), autoescape=True)
(
(
keyword_argument
loader=BaseLoader()
identifier
loader
=
=
call
BaseLoader()
identifier
BaseLoader
argument_list
()
(
(
)
)
,
,
keyword_argument
autoescape=True
identifier
autoescape
=
=
true
True
)
)
block
Clear any registered filters, tests...
comment
Clear any registered filters, tests, and glob...
expression_statement
self.env.filters.clear()
call
self.env.filters.clear()
attribute
self.env.filters.clear
attribute
self.env.filters
identifier
self
.
.
identifier
env
.
.
identifier
clear
argument_list
()
(
(
)
)
expression_statement
self.env.tests.clear()
call
self.env.tests.clear()
attribute
self.env.tests.clear
attribute
self.env.tests
identifier
self
.
.
identifier
env
.
.
identifier
clear
argument_list
()
(
(
)
)
expression_statement
self.env.globals.clear()
call
self.env.globals.clear()
attribute
self.env.globals.clear
attribute
self.env.globals
identifier
self
.
.
identifier
env
.
.
identifier
clear
argument_list
()
(
(
)
)
identifier
format_string
parameters
(self, template_str: str, values=None, **kwargs)
identifier
self
default_parameter
template_str: str, values=None
identifier
template_str
=
=
none
None
keyword_argument
kwargs
block
-> str:
For python.format compatibil...
comment
For python.format compatibility: replace all ...
comment
But avoid replacing {{...}} to {{{...}}}.
assignment
template_str = re.sub(r"(?<!{){[a-zA-Z0-9_]+}...
identifier
template_str
=
=
call
re.sub(r"(?<!{){[a-zA-Z0-9_]+}", r"{\g<0>}...
attribute
re.sub
identifier
re
.
.
identifier
sub
(
(
string
"(?<!{){[a-zA-Z0-9_]+}"
,
,
string
r"{\g<0>}"
)
)
identifier
template_str
assignment
template = self.env.from_string(template_str)
identifier
template
=
=
call
self.env.from_string(template_str)
attribute
self.env.from_string
identifier
self
.
.
identifier
env
.
.
identifier
from_string
(
(
identifier
template_str
)
)
return_statement
return template.render(values or {}, **kwargs)
call
template.render(values or {}, **kwargs)
attribute
template.render
identifier
template
.
.
identifier
render
(
(
identifier
values
or
dictionary
{ }
,
,
keyword_argument
kwargs
)
)