

# Jinja2

- язык шаблонов, использующийся в Python.

Jinja используется для генерации текстовых файлов на основе одного или нескольких шаблонов.

```
pip install Jinja
```


```
from jinja2 import Template

class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def getName(self):
        return self.name
    def getAge(self):
        return self.age

per = Person('Max', 30)

tm = Template('Мне {{p.getAge()}} лет и зовут меня {{p.getName()}}.')
msg = tm.render(p = per)
```

- **{%...%}** - спецификатор шаблона
- **{{...}}** - выражение для вставки конструкций Python в шаблон
- **{#...#}** - блок комментариев
- **#...##** - строковый комментарий
- **{{ foo.bar }}** - переменные  
**{{ foo['bar'] }}**
- **{% raw %} ... {% endraw %}** - экранирование
- **e (escape)** - экранирование спец.символов
- **{% for<выражение> -%} <повторяемый фрагмент> {% endfor -%}** - формирование списка на основе итерируемого объекта
- **{% if<условие> -%} <фрагмент при истинности условия>**  
**{% elif <условие> -%} <фрагмент при истинности условия>**  
**{% else -%} <фрагмент при ложности>**  
**{% endif -%}**  
- условие
- **{% filter <название фильтра> %} ... {% endfilter %}** - фильтры отделяются от переменной вертикальной чертой (  ) и могут иметь необязательные аргументы в круглых скобках  
**sum(iterable | sum( attribute=None, start=0))** - вычисление суммы поля коллекции

## Список встроенных фильтров

<code>abs()</code>	<code>float()</code>	<code>lower()</code>	<code>round()</code>	<code>tojson()</code>
<code>attr()</code>	<code>forceescape()</code>	<code>map()</code>	<code>safe()</code>	<code>trim()</code>
<code>batch()</code>	<code>format()</code>	<code>max()</code>	<code>select()</code>	<code>truncate()</code>
<code>capitalize()</code>	<code>groupby()</code>	<code>min()</code>	<code>selectattr()</code>	<code>unique()</code>
<code>center()</code>	<code>indent()</code>	<code>pprint()</code>	<code>slice()</code>	<code>upper()</code>
<code>default()</code>	<code>int()</code>	<code>random()</code>	<code>sort()</code>	<code>urlencode()</code>
<code>dictsort()</code>	<code>join()</code>	<code>reject()</code>	<code>string()</code>	<code>urlize()</code>
<code>escape()</code>	<code>last()</code>	<code>rejectattr()</code>	<code>striptags()</code>	<code>wordcount()</code>
<code>filesizeformat()</code>	<code>length()</code>	<code>replace()</code>	<code>sum()</code>	<code>wordwrap()</code>
<code>first()</code>	<code>list()</code>	<code>reverse()</code>	<code>title()</code>	<code>xmlattr()</code>

- `{% macro <имя(набор параметров name, value="", size=20)> %}...{% endmacro %}` - макроопределения, избежать поторений
- `{% call <имя(параметры)> вызов макроса %}...{% endcall %}` - вложенные макросы
- `{{ caller(u) }}` - вызов вложенного макроса

## Загрузчики шаблонов

- `FileSystemLoader` - для загрузки шаблонов из файловой системы нашего устройства
- `PackageLoader` - для загрузки шаблонов из пакета
- `DictLoader` - для загрузки шаблонов из словаря
- `FunctionLoader` - для загрузки на основе функции
- `PrefixLoader` - загрузчик, использующий словарь для построения подкаталогов
- `ChoiceLoader` - загрузчик, содержащий список других загрузчиков(если один не сработает, выбирается следующий)
- `ModuleLoader` - загрузчик для скомпилированных шаблонов

```
from jinja2 import Environment, FileSystemLoader

persons = [...]

file_loader = FileSystemLoader('name_file')
env = Environment(loader=file_loader)

tm = env.get_template('main.html') #Template
msg = tm.render(users = persons)
```

## Инструкции include

```
{% include 'header.html' %}
{% include 'header2.html' ignore missing %}
{% include 'main.html' %}
```

```
{% include ['article1.html', 'article2.html'] %}
{% include 'footer.html' %}
```

## Макросы

```
{% macro name_m(arg) -%}
...
{%- endmacro %}

_____

{% import 'name_m.html' as name_1 %}

{{ name_1.name_m(...) }}
```

## Наследование

- **{% extends "base.html" %}** - расширение шаблона
- **{% block footer %} ... {% endblock footer %}** - блок
- **{{ super() }}** - отображение родительского блока

## Base Template

```
<!DOCTYPE html>
<html lang="en">
<head>
  {% block head %}
  <link rel="stylesheet" href="style.css" />
  <title>{% block title %}{% endblock %} - My Webpage</title>
  {% endblock %}
</head>
<body>
  <div id="content">{% block content %}{% endblock %}</div>
  <div id="footer">
    {% block footer %}
    &copy; Copyright 2008 by <a href="http://domain.invalid/">you</a>.
    {% endblock footer %}
  </div>
</body>
</html>
```

## Child Template

```
{% extends "base.html" %} #этот шаблон «расширяет» другой шаблон
{% block title %}Index{% endblock %}
{% block head %}
  {{ super() }}
  <style type="text/css">
    .important { color: #336699; }
  </style>
{% endblock %}
{% block content %}
  <h1>Index</h1>
  <p class="important">
    Welcome to my awesome homepage.
```

```
</p>
{% endblock %}
```

**documentation:** <https://jinja.palletsprojects.com/en/2.11.x/templates/>