

Priscilla

pcms: [allegatilib](#) [allegato](#) [areetematiche](#) [countallegatilingua](#) [feedreader](#) [foto](#) [galleria](#) [getcontent](#) [getpagina](#) [getprops](#) [hreflang](#) [immaginilib](#) [jsonreader](#) [linka](#) [listagallerie](#) [loadcss](#) [lo](#)

notizie: [areenotizia](#) [notizie](#) [tipologienotizia](#)

postie: [elenconewsletter](#) [newsletteriscrizione](#)

_schedario: [schedari](#) [schedario](#) [schedarioresults](#) [schedariosearch](#)

allegatilib

Stampa una lista di allegati che hanno un dato tag

- **categoria** categoria da richiamare
- **tag** tag da richiamare
- **order** verso di ordinamento sul nome del file (asc|desc); default: asc

Almeno uno dei due parametri (tag o categoria) deve essere compilato

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% allegatilib tag=notizie-sport %}
<ul>
  {% for allegato in allegatilib %}
    <li>{{ allegato.filename }}</li>
  {% endfor %}
</ul>
```

[Index](#)

allegato

include un allegato di cui si finisce l'id

- **id** id dell'allegato da richiamare

Esempi:

```
{% allegato id=26 %}
```

[Index](#)

areetematiche

pagine che appartengono a una data area tematica

- **area** variabile che indica l'area tematica, passata ad esempio dalla pagina corrente
- **slug** slug dell'area tematica di cui si vogliono listare le pagine
- **lista** se posto a true elenca tutte le aree tematiche

Se non vengono passati parametri, stampa tutti i capo area

Permette di stampare anche i sottofilii delle pagine elencate Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

Tutte le pagine che hanno la stessa area della pagina corrente

```
{% areetematiche area=p.areatematica %}
<ul>
  {% for a in areetematiche %}
    <li><a href="{{ a.get_absolute_url }}">{{ a.titolo }}</a></li>
  {% endfor %}
</ul>
```

Passa una ben precisa area tematica tramite il parametro slug:

```
{% areetematiche slug=rosso %}
<ul>
  {% for a in areetematiche %}
    <li><a href="{{ a.get_absolute_url }}">{{ a.titolo }}</a></li>
  {% endfor %}
</ul>
```

Elenca tutte le aree tematiche:

```
{% areetematiche lista=true %}
<ul>
  {% for at in areetematiche %}
    <li>{{ at.nome }}</li>
  {% endfor %}
</ul>
```

E' possibile anche passare lo slug da un oggetto notizia in modo da mostrare tutte le pagine che hanno la stessa area tematica della notizia corrente

```
{% areetematiche slug=notizia.area.slug %}
```

Stampa dei sottofilii:

```
{% areetematiche area=p.areatematica %}
<ul>
  {% for a in areetematiche %}
    <li>
      <a href="{{ a.get_absolute_url }}">{{ a.titolo }}</a>
      {% if a.subfigli %}
        <ul>
          {% for f in a.subfigli %}
            <li><a {% if f.id == p.id %} class="corrente" {% endif %} href="{{ f.get_absolute_url }}">{{ f.titolo }}</a></li>
          {% endfor %}
        </ul>
      {% endif %}
    </li>
  {% endfor %}
</ul>
```

Stampa le pagine capo area:

```
{% areetematiche %}
<ul>
{% for a in areetematiche %}
  <li><a href="{{ a.get_absolute_url }}">{{ a.titolo }} [Area: {{ a.areatematica.nome }}]</a></li>
{% endfor %}
</ul>
```

[Index](#)

countallegatilingua

conta gli allegati di una pagina in una data lingua

- **pagina** pagina di cui si vogliono contare gli allegati
- **lingua** codice lingua

Salva il risultato in una variabile che ha lo stesso nome del tag

Gli allegati marcati come 'all' (presenti in tutte le lingue) vengono sempre contati

Esempi:

Conta gli allegati della lingua corrente

```
{% countallegatilingua pagina=p lingua=LANGUAGE_CODE %}

{% if countallegatilingua %}
  <p>Numero degli allegati in {{ LANGUAGE_CODE }}: {{ countallegatilingua }}</p>
{% endif %}
```

Conta gli allegati in lingua inglese:

```
{% countallegatilingua pagina=p lingua="en" %}
```

[Index](#)

feedreader

lettore di feed

- **url** url del feed da caricare. Eventuali caratteri '=' dovranno essere urlencoded
- **n** numero di notizie da mostrare; default: 5

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% feedreader url=http://baratto.occhiaperti.net/feed.xml n=3 %}
{% for f in feedreader %}
  <p><a href="{{ f.link }}">{{ f.title }}</a></p>
{% endfor %}
```

[Index](#)

foto

include una foto di cui si fornisce l'id

- **id** id della foto da richiamare
- **size** dimensione dell'immagine da richiamare; default: mini
- **attach** se posto a true permette di scaricare un eventuale allegato (tag a attorno al tag img) [alternativo al parametro link]
- **link** se posto a true pone il link specificato nel pannello immagine attorno al tag img [alternativo al parametro attach]
- **download** se posto a true stampa un link all'immagine e non un tag img

Esempi:

```
{% foto id=12 %}
```

Richiama un'immagine specificando la dimensione:

```
{% foto id=1 size=mini %}
```

Inserisce il link attorno al tag img:

```
{% foto id=1 link=true %}
```

Permette di scaricare un eventuale allegato:

```
{% foto id=1 attach=true size=mini %}
```

Link all'immagine:

```
{% foto id=1 download=true %}
```

E' possibile definire anche a quale dimensione immagine fare il link

```
{% foto id=1 download=true size=medium %}
```

[Index](#)

galleria

richiama le immagini di una galleria mostrando le immagini in una griglia

Parametri posizionali:

- **idgalleria** [intero] id della galleria immagini
- **nrows** [intero] numero di righe della griglia; default: 5
- **ncols** [intero] numero di colonne della griglia; default: 1
- **size** [stringa] dimensione delle immagini da richiamare; valori possibili: mini, medium, large, original; default: medium
- **plugin** [stringa] (colorbox|prettyphoto) plugin di galleria da usare: colorbox (default) o prettyphoto

Importante: Richiede jQuery e plugin Colorbox (plugin e css):

```
<script type="text/javascript" src="/static/js/jquery-ui/js/jquery-1.7.1.min.js"></script>
```

```
<script type="text/javascript" src="/static/js/colorbox/colorbox/jquery.colorbox-min.js"></script>
```

e relativo css:

```
<link rel="stylesheet" type="text/css" href="/static/js/colorbox/example1/colorbox.css"/>
```

Se si decide di usare il plugin prettyphoto, includere:

```
<script type="text/javascript" src="/static/js/prettyPhoto/jquery.prettyPhoto.js"></script>
```

e relativo css:

```
<link rel="stylesheet" type="text/css" href="/static/js/prettyPhoto/css/prettyPhoto.css" />
```

Esempi:

Con parametri di default:

```
{% galleria 1 %}
```

Con una griglia 3 x 2 e dimensione immagine a mini:

```
{% galleria 1 3 2 'mini' %}
```

Chiama il plugin prettyphoto invece del colorbox:

```
{% galleria 7 2 2 'medium' 'prettyphoto' %}
```

[Index](#)

getcontent

legge una serie di record da un contenuto fornito

- **obj** oggetto di cui si richiedono record
- **n** numero di elementi da mostrare; default: 10
- **order** campo secondo cui ordinare i risultati; default: -id

Attenzione: non verifica se il record e' pubblico o meno; perciò' per notizie e pagine e' consigliato usare i tag specifici

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% getcontent obj=pcms.Pagina %}
<ul>
{% for r in getcontent %}
  <li>{{ r }}</li>
{% endfor %}
</ul>
```

Tipo di oggetto richiamabili:

```
pcms.Pagina, pcms.Allegato, pcms.Immagine, pcms.Galleria, pcms.Menu, pcms.AreaTematica
notizie.Notizia, notizie.Area, notizie.Tipologia
postie.Newsletter, postie.Gruppo
```

[Index](#)

getpagina

restituisce un oggetto pagina da partire da un id o slug

- **id** id della pagina
- **slug** slug della pagina da richiamare
- **vslug** come il precedente ma si passa una variabile

Salva il risultato in una variabile che ha lo stesso nome del tag

Esempi:

```
{% getpagina id=1 %}
<p>La pagina ha titolo: {{ getpagina.titolo }}</p>

{% getpagina slug=pepe-mazzo %}
<p>{{ getpagina.testo|safe }}</p>

{% getpagina vslug=area.slug %}
<p>{{ getpagina.titolo }}</p>
```

[Index](#)

getprops

stampa le proprieta' (e il loro tipo) di un oggetto

- **obj** oggetto di cui si richiedono le proprieta'

Esempi:

Stampa le proprieta' (e il loro tipo) dell'oggetto di pagina **p** (l'oggetto p e' sempre disponibile in template di pagina)

```
{% getprops obj=p %}
```

Il comando produce un elenco di proprieta' dell'oggetto, ad esempio:

```
Oggetto: Pagina
id: Auto
position: Integer
titolo: Char
slug: Slug
sottotitolo: Char
testo: Text
areatematica: ForeignKey
capoarea: Boolean
menu: ForeignKey
template: FilePath
```

```
css: FilePath
figliadi: ForeignKey
redirect_to: URL
data_inizio_pubblicazione: DateTime
data_fine_pubblicazione: DateTime
pubblica: Boolean
newsletter: Boolean
is_home: Boolean
tags: Char
dc_title: Char
keywords: Char
description: Char
originalid: PositiveInteger
originalcontent: Char
originalparent: PositiveInteger
iguser: ForeignKey
created: DateTime
modified: DateTime
inothlang: Boolean
```

Per stampare nel template, ad esempio, il titolo si usera':

```
{{ p.titolo }}
```

Da notare che alcune proprieta' sono riferimento ad altri oggetti, il caso di *ForeignKey*. Per ottenerne ulteriori informazioni su questi oggetti si puo' usare lo stesso tag. Ad esempio per l'area tematica associata alla pagina:

```
{% getprops obj=p.areatematica %}
```

che produce:

```
Oggetto: AreaTematica
id: Auto
nome: Char
slug: Slug
classsecs: Char
```

[Index](#)

hreflang

stampa un elenco di url con le lingue in cui e' presente la pagina corrente

Esempi:

```
{% hreflang %}
```

Il comando precedente produce, ad esempio:

[Index](#)

immaginilist

lista le immagini presenti in una galleria di cui si fornisce il nome

- **galleria** nome della galleria di cui si vogliono listare le immagini

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% imaginilist galleria=prova-banner %}
```

Ocorre poi listare le immagini:

```
<ul>
{% for img in imaginilist %}
  <li><p>{{ img.nome }}</p></li>
{% endfor %}
</ul>
```

[Index](#)

jsonreader

lettore di json

- **url** url del json da caricare. Eventuali caratteri '=' dovranno essere urlencoded

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% jsonreader url=http://baratto.occhiaperti.net/articoli.json %}
{% for j in jsonreader %}
  {{ j.name }}
{% endfor %}
```

[Index](#)

linka

crea un link a una pagina interna di cui si fornisce l'id

- **id** id della pagina da linkare

In caso di impossibilita' di ispezionare l'oggetto fornito restituisce la stringa N/A

Esempi:

Link alla pagina avente id 2:

```
{% linka id=2 %}
```

[Index](#)

listagallerie

lista le gallerie presenti

- *n* numero delle gallerie da listare; default: 10
- *order* ordinamento da applicare ai risultati; default: -id

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

Lista le ultime 5 gallerie create:

```
{% listagallerie n=5 %}
```

e stampa un link a ognuna delle gallerie:

```
<ul>
{% for galleria in listagallerie %}
<li><a href="{{ galleria.get_absolute_url }}">{{ galleria.titolo }}</a></li>
{% endfor %}
</ul>
```

Il link manda alla pagina dettaglio della galleria. E' richiesto il template gallery.html

Ordina secondo il nome della galleria:

```
{% listagallerie order=nome %}
```

[Index](#)

loadcss

carica un file css

- *file* css da caricare
- *media* parametro opzionale da passare al tag link

stampa un tag `link` per caricare un file css aggiungendo un timestamp per riaggiornare automaticamente la cache del browser se il file viene modificato. Il percorso del file da caricare e' relativo alla directory dei file static (/static) oppure essere assoluto

Esempi:

Carica un css dalla directory dei file statici:

```
{% loadcss file="js/colorbox/example1/colorbox.css" %}
che produce
<link rel="stylesheet" type="text/css" href="/static/js/colorbox/example1/colorbox.css?1346934794" />
```

oppure, carica un css dall'omonima variabile di pagina corrente:

```
{% loadcss file=p.css %}
```

passaggio del parametro opzionale **media**:

```
{% loadcss file="css/screen.css" media="screen" %}
```

[Index](#)

loadjs

carica un file javascript

- *file* javascript da caricare

stampa un tag `script` per caricare un file javascript aggiungendo un timestamp per riaggiornare automaticamente la cache del browser se il file viene modificato. Il percorso del file da caricare e' relativo alla directory dei file statici (/static)

Esempi:

```
{% loadjs file="js/auto loader.js" %}
```

Il comando precedente produce, ad esempio:

```
<script type="text/javascript" src="/static/js/auto loader.js?1346934794"></script>
```

[Index](#)

mappa

restituisce la mappa del sito

Esempi:

```
{% mappa %}
```

[Index](#)

metagenerator

stampa il metaname generator da usare in sezione html head

Esempi:

Il tag:

```
{% metagenerator %}
```

produce:

```
<meta name="generator" content="Priscilla Framework X.Y.Z" />
```

[Index](#)

pagine

elenco di pagine per ottenere un menu o figli di una data pagina

- *menu* slug del menu di cui si vogliono ottenere le pagine
- *vmenu* variabile che contiene il nome del menu
- *figli* id della pagina di cui si vogliono listare i figli; default: homepage; si puo' indicare anche una variabile
- *correlate* id della pagina di cui si vogliono le correlate; si puo' indicare anche una variabile (es. p.id)
- *order* ordinamento da dare ai risultati; default: position; possibili valori: position, id, titolo e in generale ogni attributo dell'oggetto pagina
- *limit* limita il numero delle pagine a un valore dato

Permette di stampare anche i sottofigli

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

Stampa del menu avente slug 'menu-top':

```
{% pagine menu=menu-top order=position %}
<ul>
{% for menu in pagine %}
  <li><a href="{{ menu.get_absolute_url }}">{{ menu.titolo }}</a></li>
{% endfor %}
</ul>
```

Stampa del menu a partire da una variabile:

```
{% pagine vmenu=p.areatematica.slug %}
<ul>
{% for menu in pagine %}
  <li><a href="{{ menu.get_absolute_url }}">{{ menu.titolo }}</a></li>
{% endfor %}
</ul>
```

Pagine figlie della home (non sempre la home ha id 1, specie nel caso del multisito):

```
{% pagine figli=home %}
<ul>
{% for p in pagine %}
  <li><a href="{{ p.get_absolute_url }}">{{ p.titolo }}</a></li>
{% endfor %}
</ul>
```

Pagine figlie della pagina avente id 4 (si ordina per campo titolo inverso):

```
{% pagine figli=4 order=-titolo %}
<ul>
{% for p in pagine %}
  <li><a href="{{ p.get_absolute_url }}">{{ p.titolo }}</a></li>
{% endfor %}
</ul>
```

Pagine figlie della pagina corrente, l'id si prende dall'oggetto di pagina **p**:

```
{% pagine figli=p.id %}
<ul>
{% for p in pagine %}
  <li><a href="{{ p.get_absolute_url }}">{{ p.titolo }}</a></li>
{% endfor %}
</ul>
```

Pagine sorelle della pagina corrente, l'id si prende dall'oggetto di pagina **p**. Si passa l'id del genitore:

```
{% pagine figli=p.figliadi.id %}
<ul>
{% for s in pagine %}
  <li><a href="{{ s.get_absolute_url }}">{{ s.titolo }}</a></li>
{% endfor %}
</ul>
```

Stampa l'elenco delle ultime 10 pagine modificate:

```
{% pagine order=-modified limit=10 %}
<ul>
{% for p in pagine %}
  <li><a href="{{ p.get_absolute_url }}">{{ p.titolo }} [modificata il {{ p.modified|date:"d-m-Y" }}]</a></li>
{% endfor %}
</ul>
```

Stampa le figlie della pagina corrente ed espande i sottofigli:

```
{% pagine figli=p.id %}
<ul>
{% for p in pagine %}
  <li><a href="{{ p.get_absolute_url }}">{{ p.titolo }}</a></li>
  {% if p.subfigli %}
  <ul>
    {% for f in p.subfigli %}
      <li><a {% if f.id == p.id %} class="corrente" {% endif %} href="{{ f.get_absolute_url }}">{{ f.titolo }}</a></li>
    {% endfor %}
  </ul>
  {% endif %}
{% endfor %}
</ul>
```

Stampa le immagini correlate alla pagina:

```
{% if p.immagine_set.all %}
<ul>
  {% for imago in p.immagine_set.all %}
    <li></li>
  {% endfor %}
</ul>
{% endif %}
```

Stampa gli allegati correlati alla pagina:

```
{% if p.allegato_set.all %}
<ul>
  {% for allegato in p.allegato_set.all %}
    <li><a href="{{ allegato.filename.url }}">{{ allegato.descrizione }}</a></li>
  {% endfor %}
</ul>
```

```
</ul>
{% endif %}
```

Stampa le immagini delle gallerie correlate alla pagina:

```
{% if p.gallerie.all %}
  {% for galleria in p.gallerie.all %}
    <ul id="galleria-{{ galleria.id }}">
      {% if galleria.immagine_set.all %}
        {% for imago in galleria.immagine_set.all %}
          <li></li>
        {% endfor %}
      {% endif %}
    </ul>
  {% endfor %}
{% endif %}
```

Stampa i codici video associati alla pagina:

```
{% if p.video_set.all %}
  <ul>
    {% for video in p.video_set.all %}
      <li>{{ video.videoencode }}</li>
    {% endfor %}
  </ul>
{% endif %}
```

Stampa le pagine correlate a quella corrente

```
{% pagine correlate=p.id %}

<ul>
  {% for p in pagine %}
    <li>{{ p.titolo }}</li>
  {% endfor %}
</ul>
```

[Index](#)

pollicino

stampa le briciole di pane per arrivare alla pagina in cui e' inserito

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% pollicino %}
{% for briciola in pollicino %}
  &gt; <a href="{% if briciola.is_home %}/{% else %}{{ briciola.get_absolute_url }}{% endif %}">{{ briciola.titolo }}</a>
{% endfor %}
```

[Index](#)

scroller

richiama le immagini di una galleria applicando un effetto di scrolling

Parametri posizionali:

- **idgalleria** [intero] id della galleria immagini
- **size** [stringa] dimensione delle immagini da richiamare; valori possibili: mini, medium, large, original; default: medium

Esempi:

```
{% scroller 3 %}
{% scroller 3 'large' %}
```

Importante: occorre includere librerie javascript:

```
<script type="text/javascript" src="/static/js/scroll/jquery.scrollTo-1.4.3.1-min.js"></script>
<script type="text/javascript" src="/static/js/scroll/jquery.serialScroll-1.2.2-min.js"></script>
<script type="text/javascript" src="/static/js/scroll/scroll-init.js"></script>
```

e opportuni stili css, ad esempio:

```
<link rel="stylesheet" type="text/css" href="/static/js/scroll/scroller.css" />
```

Richiede jQuery.

[Index](#)

slide

richiama le immagini di una galleria applicando un effetto slide

Parametri posizionali:

- **idgalleria** [intero] id della galleria immagini
- **size** [stringa] dimensione delle immagini da richiamare; valori possibili: mini, medium, large, original; default: original
- **slidetime** [intero] numero di millisecondi tra le rotazioni; default: 5000 Esempi:

```
{% slide 2 %}
```

```
{% slide 2 'medium' %}
```

Importante: richiede JQuery e l'inclusione di stili css applicati alla classe slideshow oppure all'id dello slideshow:

```
<style>
.slideshow {
  position:relative;
  height:350px;
}

.slideshow IMG {
  position:absolute;
  top:0;
  left:0;
```

```

    z-index:8;
  }

  .slideshow IMG.active {
    z-index:10;
  }

  .slideshow IMG.last-active {
    z-index:9;
  }
}
</style>

```

[Index](#)

thebanner

richiama le immagini di una galleria come banner

Parametri posizionali:

- **idgalleria** [intero] id della galleria immagini
- **size** [stringa] dimensione delle immagini da richiamare; valori possibili: mini, medium, large, original; default: original
- **blocklink** [stringa] se impostato (diverso da "") i banner non sono cliccabili
- **nimage** [intero] numero di immagini da richiamare; default: 4

Esempi:

Richiama i banner dalla galleria avente id 1:

```
{% thebanner 1 %}
```

fornisce la dimensione delle immagini:

```
{% thebanner 1 'medium' %}
```

rende le immagini non cliccabili:

```
{% thebanner 3 'medium' 'false' %}
```

lascia le immagini cliccabili, ma ne richiama solo una:

```
{% thebanner 3 'medium' '' 1 %}
```

[Index](#)

wookmark

richiama le immagini di una galleria applicando un effetto alla Wookmark

Parametri posizionali:

- **idgalleria** [intero] id della galleria immagini
- **nimage** [intero] numero di immagini da mostrare inizialmente; default=9
- **cszize** [stringa] dimensione delle immagini da richiamare; valori possibili: mini, medium, large, original; default: medium

Esempi:

```
{% wookmark 3 4 %}
```

Richiede jQuery.

[Index](#)

areenotizia

Stampa l'elenco delle aree delle notizie

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```

{% areenotizia %}
<ul>
{% for a in areenotizia %}
  <li>{{ a.nome }}</li>
{% endfor %}
</ul>

```

[Index](#)

notizie

Stampa l'elenco delle notizie

- **area** area della notizia da richiamare; si fornisce lo slug dell'area
- **varea** sempre slug area della notizia, ma da una variabile
- **altrearee** (includi | soltanto): utilizzato con opzione area o varea: cerca l'area anche in altre aree oltre che nell'area principale (includi) oppure solo nelle altre aree (soltanto)
- **primopiano** se impostato (primopiano=true) stampa le notizie che sono state marcate come Primo piano
- **_home_** se impostato (home=true) stampa le notizie che son state marcate come Mostra in home page
- **tipologia** filtra di una notizia di una data tipologia
- **vtipologia** sempre slug tipologia, ma da una variabile
- **excludearea** esclude una data area notizie dai risultati (o piu' aree dando piu' slug separati da ,)
- **excludetipologia** esclude una data tipologia dai risultati (o piu' tipologie dando slug separati da ,)
- **order** ordina secondo un dato campo (default: -datanotizia)
- **nold** se impostato (nold=true) non include le notizie del passato
- **n** numero massimo di notizie da mostrare; default: 10
- **year** richiama le notizie di un dato anno; year=current per l'anno corrente

Salva i risultati in una variabile che ha lo stesso nome del tag (notizie)

Esempi:

Notizie di una data area:

```

{% notizie area=cavallo %}
<ul>
{% for n in notizie %}

```

```
<li>{{ n.datanotizia }} * {{ n.titolo }}</li>
{% endfor %}
</ul>
```

Si richiama sempre un'area, ma invece di definirla si ottiene dinamicamente da una variabile; mostra anche la prima di eventuali immagini associate alla notizia:

```
{% notizie varea=p.areatematica.slug %}
<ul>
{% for n in notizie %}
  <li>
    {{ n.datanotizia }} * {{ n.titolo }}
    {% if n.immagine_set.all %}
      
    {% endif %}
  </li>
{% endfor %}
</ul>
```

Notizia di una data area, cercando anche nelle aree secondarie:

```
{% notizie area=acquisti altrearee=includi %}
<ul>
{% for n in notizie %}
  <li><a href="{{ n.get_absolute_url }}">{{ n.titolo }}</a></li>
{% endfor %}
</ul>
```

Notizie di una data area, cercando solo nelle aree secondarie:

```
{% notizie area=acquisti altrearee=soltanto %}
<ul>
{% for n in notizie %}
  <li><a href="{{ n.get_absolute_url }}">{{ n.titolo }}</a></li>
{% endfor %}
</ul>
```

Solo le notizie marcate come in primo piano:

```
{% notizie primopiano=true %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}{% endif %}</li>
{% endfor %}
</ul>
```

Solo le notizie marcate per essere mostrate in home:

```
{% notizie home=true %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}{% endif %}</li>
{% endfor %}
</ul>
```

Richiamo per tipologia:

```
{% notizie tipologia=elba %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}</li>
{% endfor %}
</ul>
```

Con possibilita' di escludere un'area:

```
{% notizie tipologia=elba escludearea=apnea %}
```

O piu' aree:

```
{% notizie tipologia=elba escludearea=apnea,nuoto %}
```

Richiamo di area e tipologia, limita il numero delle notizie:

```
{% notizie area=cavallo tipologia=elba n=1 %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}</li>
{% endfor %}
</ul>
```

Nel caso di una notizia di primo piano (n=1) si puo' accedere all'immagine principale tramite mainpic

```
{% notizie n=1 %}
<ul>
{% for n in notizie %}
  <li>{{ n.titolo }}
    {% if n.mainpic %}
      
    {% endif %}
  </li>
{% endfor %}
</ul>
```

Ultime notizie:

```
{% notizie n=5 %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}</li>
{% endfor %}
</ul>
```

Non mostra le notizie del passato:

```
{% notizie n=5 nold=true %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}</li>
</ul>
```

```
{% endfor %}
</ul>
```

Mostra le notizie dell'anno corrente

```
{% notizie year=current %}
<ul>
{% for n in notizie %}
  <li>{{ n.datanotizia }} * {{ n.titolo }}</li>
{% endfor %}
</ul>
```

[Index](#)

tipologienotizia

Stampa l'elenco delle tipologie delle notizie

Salva i risultati in una variabile che ha lo stesso nome del tag

Esempi:

```
{% tipologienotizia %}
<ul>
{% for t in tipologienotizia %}
  <li>{{ t.nome }}</li>
{% endfor %}
</ul>
```

[Index](#)

elenconewsletter

stampa l'elenco delle newsletter che sono state spedite

- *area* area della newsletter da richiamare; si fornisce lo slug dell'area
- *varea* sempre slug area della newsletter, ma da una variabile
- *user* filtra le newsletter secondo un dato username
- *vuser* sempre username, ma da una variabile
- *n* numero di newsletter da mostrare; default: 10
- *order* ordinamento; default: -date_sent

Esempi:

Elenco delle ultime 10 newsletter spedite

```
{% elenconewsletter n=10 %}
<ul>
{% for n in elenconewsletter %}
  <li><a href="{{ n.get_absolute_url }}">{{ n.titolo }}</a></li>
{% endfor %}
</ul>
```

Elenco delle ultime 10 newsletter dell'area di cui si fornisce lo slug news

```
{% elenconewsletter area=news n=10 %}
<ul>
{% for n in elenconewsletter %}
  <li><a href="{{ n.get_absolute_url }}">{{ n.titolo }}</a></li>
{% endfor %}
</ul>
```

Elenco delle ultime 10 newsletter con area impostata da variabile (ad esempio l'area della notizia corrente)

```
{% elenconewsletter varea=newsletter.area n=10 %}
<ul>
{% for n in elenconewsletter %}
  <li><a href="{{ n.get_absolute_url }}">{{ n.titolo }}</a></li>
{% endfor %}
</ul>
```

Elenco delle newsletter di un dato utente:

```
{% elenconewsletter user=username n=10 %}
<ul>
{% for n in elenconewsletter %}
  <li><a href="{{ n.get_absolute_url }}">{{ n.titolo }}</a></li>
{% endfor %}
</ul>
```

Per prendere, ad esempio, lo username dal proprietario della pagina corrente

```
{% elenconewsletter vuser=p.iguser n=10 %}
```

[Index](#)

newsletteriscrizione

stampa un form di iscrizione alla newsletter

- *idgruppi* id dei gruppi da mostrare in checkbox; default tutti i gruppi pubblici
- *testomail* testo da mostrare all'interno dell'input email; default: qui la tua mail
- *testainvio* testo del bottone di iscrizione; default: Iscrivimi
- *testourlprivacy* testo del link che porta alla pagina privacy; default: Confermo di aver letto e accetto l'informativa sulla privacy
- *urlprivacy* url che rimanda all'informativa sulla privacy; default: /
- *msgrequiredprivacy* messaggio per richiedere il check privacy; default: Occorre accettare l'informativa sulla privacy"
- *formid* id del form della newsletter; default: postie-newsletter

Esempi:

Form di iscrizione alla newsletter con scelta di tutti i gruppi pubblici:

```
{% newsletteriscrizione %}
```

Solo alcuni gruppi di cui si indicano gli id separati da virgola; i gruppi debbono essere a iscrizione pubblica

```
{% newsletteriscrizione idgruppi=2,3,7 %}
```

Con personalizzazione di etichette:

```
{% newsletteriscrizione testomail="Inserisci qui la tua email" testoinvio="Iscrivimi alla newsletter" %}
```

[Index](#)

shedari

lista gli shedari

- *categoria* mostra solo gli shedari appartenenti alla categoria di cui si fornisce l'id

Esempi:

```
{% shedari %}
```

Mostra gli shedari appartenenti alla categoria avente id 2

```
{% shedari categoria=2 %}
```

Per stampare l'elenco dei link agli shedari:

```
{% for s in shedari %} {{ s.nome }}  
{% endfor %}
```

Per inserire nel link anche il campo secondo cui ordinare: aggiungere il nome del campo a fine link.

Ordina secondo nome inverso: `{{ s.nome }}`

Ordina secondo il campo numerico order: `{{ s.nome }}`

[Index](#)

shedario

Mostra uno shedario fornendo l'id

Parametri posizionali:

- *idschedario* [intero] id dello shedario
- *order* [stringa] campo secondo cui ordinare le schede (default: nome)
- *template* [stringa] template dello shedario in alternativa a quello di default: scheda.html (deve essere nella cartella templates/schedario/)

Esempi:

Con parametri di default dove 1 e' l'id dello shedario:

```
{% shedario 1 %}
```

Ordina le schede secondo il campo numerico:

```
{% shedario 1 "order" %}
```

Ordina le schede per nome inverso:

```
{% shedario 1 "-nome" %}
```

Passando anche il nome del template (deve essere nella cartella templates/schedario/):

```
{% shedario 1 "nome" "scheda_custom.html" %}
```

[Index](#)

shedarioresults

Mostra i risultati di ricerca sullo shedario

Parametri posizionali:

- *idschedario* [intero] id dello shedario in cui cercare
- *_idschedari_* [stringa] id degli shedari in cui cercare separati da virgola: in alternativa al passaggio di un id shedario singolo
- *rpp* [intero] numero di risultati da mostrare per pagina
- *template* [stringa] template di ricerca shedario in alternativa a quello di default: risultati_schedario.html (deve essere nella cartella templates/schedario/)

Esempi:

Cerca sullo shedario avente id 1:

```
{% shedarioresults 1 %}
```

Ricerca in piu' shedari. Racchiudere gli id separati da virgole tra virgolette doppie:

```
{% shedarioresults "1,4,8" %}
```

Indica il numero di risultati per pagina: 30

```
{% shedarioresults 4 30 %}
```

Passando anche il nome del template (deve essere nella cartella templates/schedario/):

```
{% shedarioresults 1 25 risultati_schedario_custom.html %}
```

[Index](#)

shedariosearch

Mostra un form di ricerca sugli shedari

Parametri posizionali:

- id della pagina che contiene il tag per mostrare i risultati: se non specificato si intende pagina corrente
- *template* [stringa] template di ricerca shedario in alternativa a quello di default: ricerca_schedario.html (deve essere nella cartella templates/schedario/)

Esempi:

Il tag per mostrare i risultati (schedarioresults) e' nella pagina avente id 12:

```
{% schedariosearch 12 %}
```

Passando anche il nome del template (deve essere nella cartella templates/schedario/:

```
{% schedariosearch 12 "search_custom.html" %}
```