

# Da Google\_Docs a Read\_the\_Docs

unknown

01 dic 2023



<b>1</b>	<b>Vantaggi dell'uso di "Read the Docs"</b>	<b>3</b>
<b>2</b>	<b>Changelog</b>	<b>5</b>
<b>3</b>	<b>L'utilità di GGeditor per i progetti di documentazione online</b>	<b>11</b>
<b>4</b>	<b>Le principali funzioni e punti di forza di GGeditor</b>	<b>13</b>
4.1	Caratteristiche di GGeditor . . . . .	14
4.2	Tutorial sull'uso di GGeditor . . . . .	17
4.3	Le funzioni su Google doc . . . . .	19
4.4	Inserimento di tabelle nel documento . . . . .	28
4.5	Inserimento di video, immagini e embedding nel documento . . . . .	30
4.6	Il lavoro da fare su Github . . . . .	32
4.7	Il lavoro da fare su Read the Docs . . . . .	43
4.8	User-guide (gli step operativi) . . . . .	45
4.9	Strumenti per i più esperti . . . . .	48
4.10	Hypothes.is per partecipare ad un documento su Read the Docs . . . . .	49
4.11	Sintassi del linguaggio reStructuredText (RST) . . . . .	56
4.12	Sintassi del linguaggio Markdown (MD) per le finalità di 'Read the Docs' . . . . .	57
4.13	Pubblicare con il design di Docs Italia . . . . .	58
4.14	Licenza, chi sono e ringraziamenti . . . . .	62
4.15	Opendatasicilia . . . . .	65



Le spiegazioni contenute in questo tutorial rappresentano una via per migliorare la qualità dei documenti pubblicati sul web e sono la traduzione in italiano del . Il tutorial spiega come usare lo strumento di **Google doc**, ed un componente aggiuntivo, per pubblicare documenti con lo stile di **Read the Docs**, utilizzando anche **GitHub**.

Questo documento nasce subito dopo la produzione del tutorial "".

---

**Per questo tutorial i seguenti ringraziamenti particolari**

Ringraziare queste persone è importante, perché con i loro approfondimenti hanno permesso di ampliare le possibilità d'uso di Read the Docs come piattaforma di pubblicazione documentale:

.....

---



---

## Vantaggi dell'uso di "Read the Docs"

---

L'uso di Read the Docs come piattaforma di pubblicazione di documenti online ha i seguenti vantaggi rispetto al formato *PDF*:

Per i nostalgici e dipendenti da documenti in formato *PDF* (non accessibili comodamente da dispositivi mobili), la documentazione esposta su Read the Docs permette di scaricare il contenuto dell'intero documento pubblicato online sia in formato *PDF* che *EPUB* oltre che *HTML*.



#### Registro di tutti gli aggiornamenti | modifiche principali apportate al progetto

---

##### Aggiornamento N. 1 - gennaio\_2020

Su Google doc cliccando su “*installa componente aggiuntivo*” non trovate più “**GGeditor**” neanche sul Marketplace di Google (capita dall’agosto 2019 per una ridefinizione dei termini d’uso di Google). Passate alla procedura del successivo aggiornamento.

---

##### Aggiornamento N.2 - 14 febbraio\_2020

In alternativa all’installazione del componente aggiuntivo su Google doc (dall’elenco dei componenti aggiuntivi forniti da Google), le funzioni svolte da GGeditor possono anche essere assicurate creando un Google doc che contiene uno script con il codice sorgente del componente **GGeditor**. Una volta creato lo script (con la procedura di seguito illustrata) avviando il comando “Commit to Github” (percorso: componenti aggiuntivi / GGeditor / Commit to Github), è possibile creare automaticamente un file in formato .RST su Github partendo dal contenuto editato su Google doc.

↓

##### I file contenuti nello script da creare su Google doc

Lo script è costituito dai seguenti files che si trovano dentro il repository :

- conversion.html
- explicitMarkup.html
- generator.gs
- github.html
- properties.gs

- reSTMetadata.gs
- settings.html
- sidebar.html
- .gs ( in cinese significa *codice*)

Per creare lo script su Google doc, andare su **menu strumenti / < > editor di script**. Nella pagina dello script copiare il codice dei 9 file di cui al repository dando lo stesso nome dei 9 file di cui sopra. Allo script così creato dare il nome GGeditor.

---

---

### Aggiornamento N.3 - 19 febbraio 2020

Messaggio “**Bad Credential**” (“Credenziali errate di Github”) su GGeditor. Github ha deprecato la sua API di autenticazione per “*nome utente*” e “*password*”, che è la causa principale del problema del messaggio “Bad Credential” («Credenziali non valide»).

, basta sostituire la password con cui si entra nell’account Github con il «**token di accesso personale**» quando si esegue il commit in GGEditor nel Google doc. I passi da seguire sono i seguenti:

1. Vai alla e **crea un token di accesso personale** in Github.com (). Quindi copia il token di accesso personale negli appunti.
2. Apri un documento Google e rimuovi tutte le credenziali archiviate precedentemente in GGEditor, quindi aggiungi un nuovo account Github con il token copiato come password.

Guarda i passi da compiere nelle schermate di Github:

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Generate new token

Select scope

Scopes determine the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations

Generate token

#### Aggiornamento N. 4 - ottobre\_2021

Messaggio di “non compilazione” del progetto su readthedocs.org e soluzione (news di fine ottobre 2021). Guarda anche la relativa .

Da fine ottobre 2021 su readthedocs.org compare una non compilazione del progetto (**build failed**) legata alla versione di Sphinx. La soluzione è inserire nel file `requirements.txt` la seguente sintassi:

```
docutils<0.18
```

come di seguito illustrato.

1- avere un denominato `requirements.txt` con il seguente codice:

```
1 sphinx-rtd-theme
2 sphinx
3 recommonmark
4 markdown
5 sphinx-markdown-tables
6 docutils<0.18
```

2- avere un `.readthedocs.yaml` con il seguente codice:

```
1 version: 2
2 python:
3   install:
4     - requirements: requirements.txt
```

---

### Aggiornamento N. 5 - maggio\_2022

Fortunatamente il *componente aggiuntivo* per Google Doc GG editor è di nuovo disponibile nello store di Google per l'installazione. Vai al seguente link:

---

---

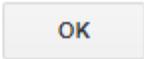
### Messaggio di errore “undefined”, soluzione

Nelle azioni di commit sul componente GGeditor potrebbe comparire un messaggio di “**undefined**” ed un messaggio di error in lingua taiwanese.

Ho aperto una issue () sul suo progetto Github e Hsin Yuan Yeh (autore di GGeditor) ha provveduto ad analizzare tempestivamente il problema ed ha fornito immediatamente una soluzione che è stata implementata nel codice sorgente di **GGeditor**.

Praticamente se dovesse capitare di leggere un messaggio “undefined” o un messaggio in lingua taiwanese del tipo ↓

無法讀取 null 的  
[ 19rZn4reioTxYQL2dLiVbxOHRMnw2llq\_5r6xtLpPqVQ ] 屬性。



La cosa da fare è:

andare in “**Componenti aggiuntivi**”, poi “**GGeditor**” e quindi su “**Setting**” dove si trova un tasto rosso di “**Reset**” che cancella tutte le informazioni di collegamento agli account Github e ai relativi file nel repository.

## Settings



The "Reset" button will clean up all your stored information of accounts and bindings.



Tutto ritorna in ordine e funziona correttamente nella procedura di Commit da Google Doc a Github. Ovviamente bisogna ricollegare il componente aggiuntivo GGeditor all'account di Github per poter continuare a effettuare i commits. Grazie Hsin Yuan Yeh per questa tempestiva soluzione al problema.



---

### L'utilità di GGeditor per i progetti di documentazione online

---

Il componente aggiuntivo **GGeditor** rappresenta uno strumento molto utile e comodo in quanto i servizi di Google Drive oggi sono molto usati anche nelle Pubbliche Amministrazioni, oltre che dai privati, per la facilità d'uso e per la funzionalità di condivisione dei documenti in gruppo.

Il lavoro principale che svolge il componente aggiuntivo **GGeditor** è quello di trasformare il testo editato su un foglio di Google doc in un file con linguaggio `.RST` dentro il repository di **Github**. Github a sua volta permette la compilazione automatica dello stesso documento su **Read the Docs** in pagine HTML. Sembra una cosa difficile nella descrizione, ma se lo faccio lo possono fare tutti, con un pizzico di pazienza e curiosità.



---

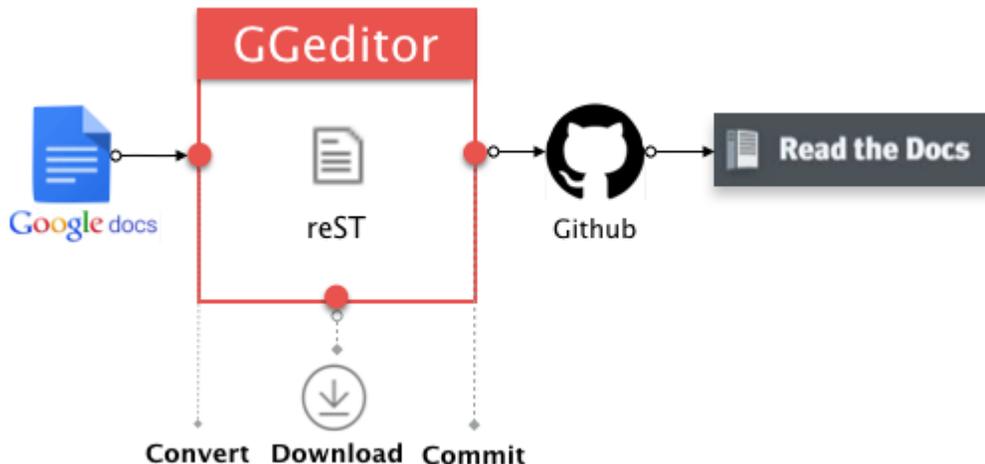
### Le principali funzioni e punti di forza di GGeditor

---

- Facile inizio per chi non ha dimestichezza con i file RST, anche per chi non ha idea dei marcatori di RST.
- Alimentato da Google Docs. Quasi la totalità di quello che vedi su Google Docs è quello che ottieni su Read the Docs. Lo stesso è per l'intero gruppo di lavoro.
- Un click per commissionare il lavoro sul repository di Github.
- Puoi vedere in anteprima il file RST generato dall'interno di Google Docs e scaricarlo nel tuo PC.
- Supporta headings, bold, italic, hyperlink, subscript e superscript.
- Supporta note a margine, immagini, liste di articolo e tabelle.
- Supporta caratteri a larghezza intera (CKJ) nelle intestazioni e nelle tabelle.
- Supporta i link interni ai bookmarks, headings e le Google Docs tabelle native di contenuti (in document table of contents).
- Supporta i link relativi ai file RST generati dai Google Docs all'interno della stessa directory e sotto-directory Google Docs.
- Supporta la tabella dei contenuti (cross-document table of content (`.. toctree: :)`) per fare generare l'indice a Read The Docs.
- Supporta tutti gli stili di «admonitions» di Read The Docs.
- Supporta account multipli per compilare i file nei repository di diversi account Github.
- Supporta la conversione di tabelle con i tags HTML to let look-and-feel e la stessa cosa è possibile per i blogger.

## 4.1 Caratteristiche di GGeditor

### 4.1.1 Da Google Doc a Read the Docs



Oggi il formato PDF rappresenta il principale formato di testo per la pubblicazione di documenti sia da parte della Pubblica Amministrazione che dai soggetti privati.

1. Purtroppo il PDF è un formato che non si adatta ai display piccoli dei dispositivi mobili, ed oggi la fruizione dei contenuti del web è molto consistente sui dispositivi mobili.
2. In più il formato PDF non consente un agevole e facile ricerca di parole all'interno del documento.

Queste due criticità lo rendono un formato ormai vetusto nel 2018, non più rispondente alle esigenze di fruibilità di contenuti documentali su display di dimensioni contenute (smartphone, tablet).

GGeditor soddisfa questi requisiti rendendo la fruizione di un documento online un'esperienza gradevole.

---

**Importante:** Questa pagina che state leggendo deriva direttamente da

---

---

**Nota:** GGeditor consente di compilare documenti sulla piattaforma di repository codice Github, che a sua volta serve per illustrare i documenti su Read the Docs in maniera gradevole, e strutturata per la fruizione dei contenuti indicizzati ad albero (capitoli, paragrafi, sotto-paragrafi).

---

#### 4.1.2 GEditor ed i file formato .rst



[il plugin GEditor installabile da Google Docs (cercalo nei «Componenti aggiuntivi»)]

---

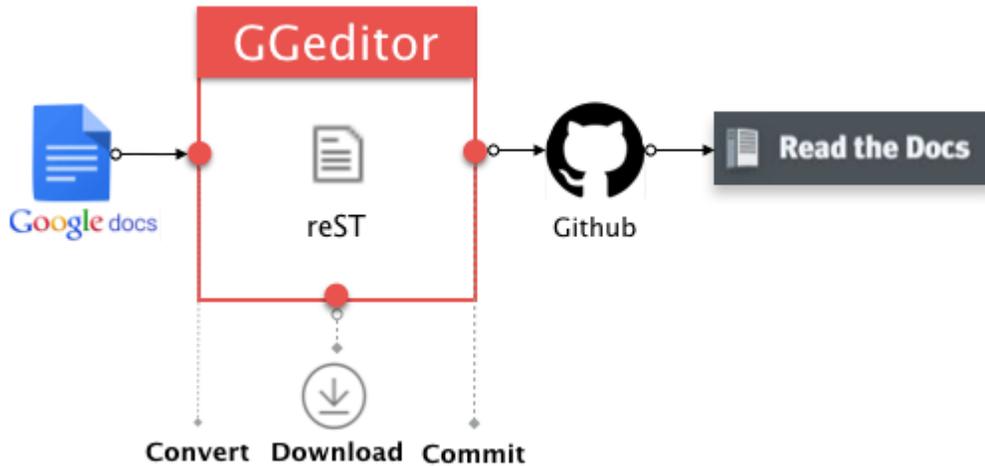
**Importante:** è un plug-in di Google Docs, creato dal taiwanese Hsin Yuan Yeh, che serve a generare file **RST** () direttamente da Google Docs.

Il file RST generato può essere compilato nel repository di Github direttamente dall'editor GG. La documentazione così creata su Google Docs può essere ospitata da venendo aggiornata automaticamente ad ogni aggiornamento del Google Docs

---

#### 4.1.3 Video tutorial di funzionamento di GEditor

#### 4.1.4 Il processo che svolge GGeditor



[Questo è il processo svolto da GGeditor: da Google Docs a GGeditor a Github a Readthedocs]

da



Google Docs

Scrivi facilmente testo in un documento senza conoscere il linguaggio RST)

a ↓



GGeditor

GGeditor è un plug-in di Google Docs che automatizza il lavoro di compilazione sul repository di Github

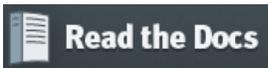
a ↓



Github

Il progetto sul repository di Github è fondamentale per esporre il documento da pubblicare su Read the Docs

a ↓



Readthedocs

Read the Docs è la piattaforma che espone documenti con un efficace architettura dei contenuti, in un formato usabile da tutte le dimensioni di display e che permette una facile ricerca di parole nel testo

### 4.1.5 I file che GGedito<sup>r</sup> genera automaticamente su Github

 cirospat	Google Docs reST Editor@2018-3-28 23:44:30
..	
 static	Create theme_overrides.css
 1.rst	Google Docs reST Editor@2018-3-28 23:44:30
 conf.py	Update conf.py
 index.rst	Google Docs reST Editor@2018-3-28 23:37:4

[immagine del repository di Github che mostra come i file RST vengono generati direttamente dall'interno di Google Docs tramite il plugin GGedito<sup>r</sup>]

## 4.2 Tutorial sull'uso di GGedito<sup>r</sup>

il tutorial è molto semplice nei passi da seguire, partendo innanzitutto da questo video:

### 4.2.1 Come installare GGedito<sup>r</sup>

, oppure in un documento Google:

1. Sul menu «componenti aggiuntivi» clicca “Installa componenti aggiuntivi”
2. Nel box di ricerca edita GGedito<sup>r</sup>, e clicca sull'icona di GGedito<sup>r</sup> per installare.

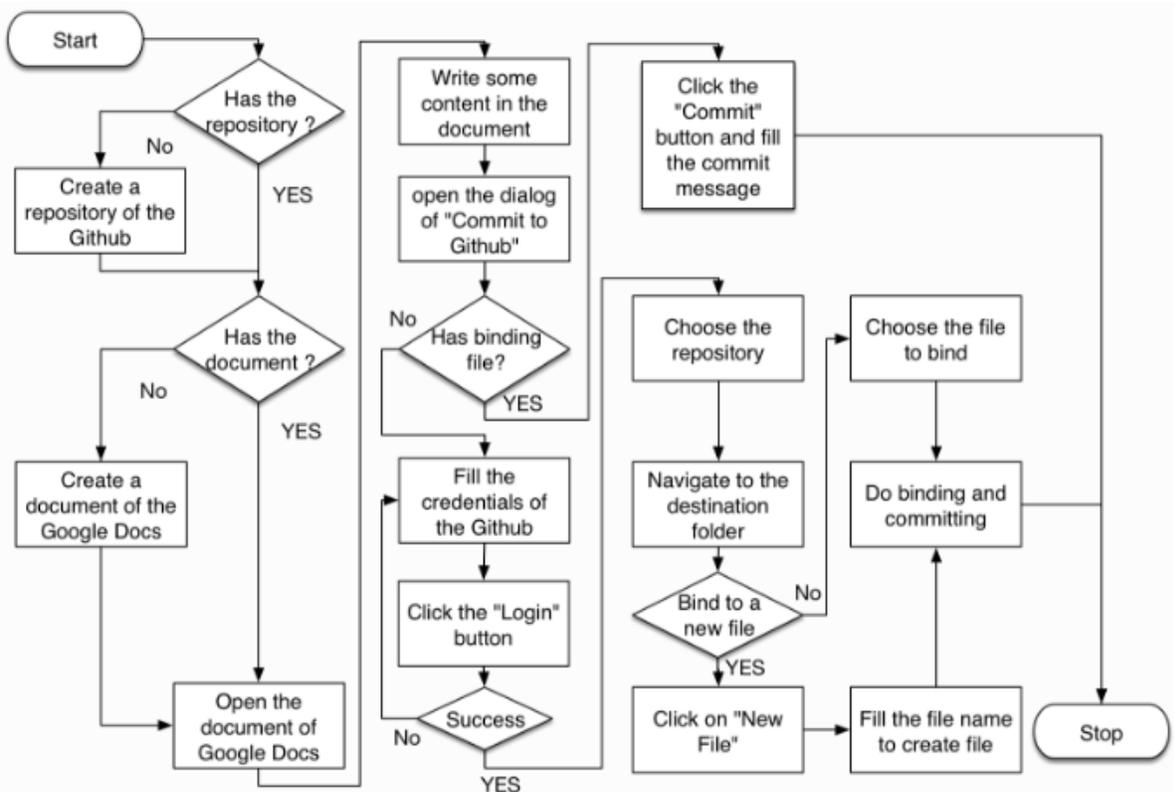
L'editor GG parte da uno scenario composto da 2 situazioni:

1. Tu già hai un repository di progetto su **Github**,
2. Tu già hai un progetto su **Readthedocs.org** che è in diretta relazione al repository su Github.

Github e Readthedocs accettano formati RST o Markup. Hai bisogno di costruire la documentazione in uno di questi due formati.

## 4.2.2 Il flusso di lavoro del processo che gestisce GEditor

Questo è il flusso di lavoro con GEditor per costruire la documentazione su Github:



## 4.2.3 Le azioni svolte da GEditor

La sequenza di azioni che vengono effettuate da GEditor su Google Doc nel processo che esso stesso gestisce. Ecco le fasi:

1. iniziare creando un nuovo documento su Google Docs.
2. Il nuovo file sarà nominato «Tutorial» (come nel caso di questa pagina che state leggendo). Il file contiene un'intestazione, un'immagine e una nota creata dal sidebar di GEditor.
3. Il nuovo file sarà compilato da GEditor sul repository del progetto Github.
4. Siccome questo è un file nuovo, un processo sarà avviato per costruire il legame agli altri documenti (es. index) nel repository Github. Il processo avviato include:
  1. log-in all'account Github che ospita il progetto su Github,
  2. azioni di navigazione dentro Github,
  3. creazione di un nuovo file
  4. e l'attività di compilazione (commit di Github).

---

**Importante:**

1. Quando nomini il tuo documento Google Doc, il nome del documento non necessita del suffisso `.rst`.
  2. Per legare il file del Google Doc sul repository del progetto (Github), Github ha bisogno del suffisso `.rst`. Il suffisso sarà automaticamente creato da GGeditor. Se nomini manualmente tu il file su Github allora aggiungi il suffisso `.rst` (sulla directory docs di Github).
- 

---

**Suggerimento:** Il contenuto di questa pagina che stai leggendo è in

---

## 4.3 Le funzioni su Google doc

### 4.3.1 Uno strumento per i principianti dei file 'rST'

Se sei all'inizio della conoscenza dei file con sintassi RST (reStructuredText) e ti senti un po' confuso sul come inserire i tuoi documenti sul portale di Read the Docs, questo strumento fa sicuramente per te. Con il componente aggiuntivo GGeditor per Google doc e questo piccolo tutorial tradotto in italiano dal tutorial originale, spero che potrai essere assistito nel realizzare il tuo lavoro in maniera facile e veloce. Se dopo averlo letto ti è rimasto qualche dubbio, beh.. contattami: .

### 4.3.2 Il plug-in di Google Doc, GGeditor

Per prima cosa da Google Doc si va su «Componenti aggiuntivi», si cerca e si installa il plug-in GGeditor



così si può cominciare ad usare il set di strumenti del menu che appare se clicchiamo su «componenti aggiuntivi», poi su GGeditor e poi ancora su «Show Markup Panel».

### 4.3.3 I settings dell'account Github

#### Settings

Accounts	Document	About
----------	----------	-------

Type	Username	Password	Show password
Github		*****	Delete

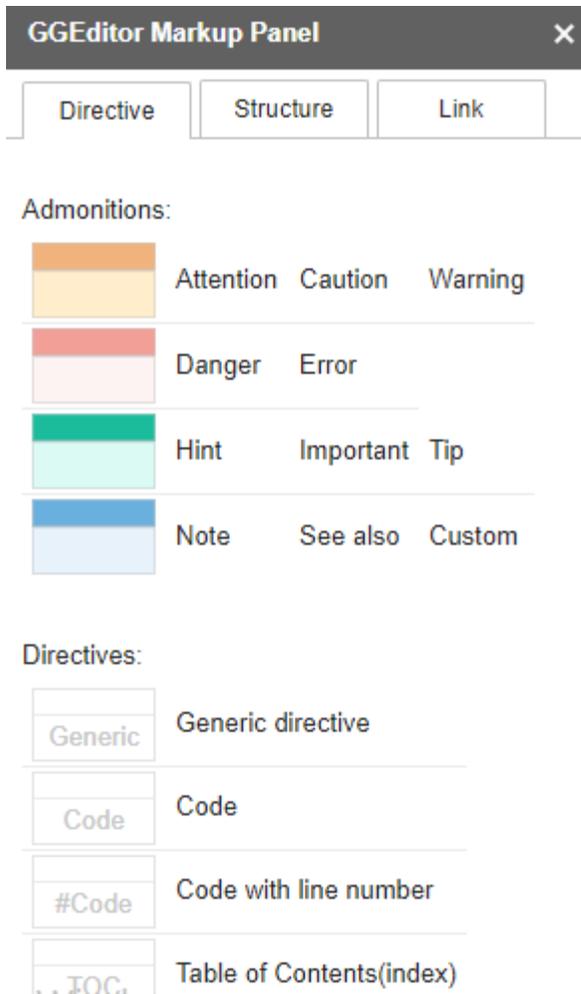
[+ Add Account](#)

Dalla finestra “settings” è possibile agganciare il proprio account Github al plugin GEditor in maniera tale che ogni volta che vogliamo commissionare un Google Doc su Github, cliccando su “Commit to Github” è possibile visualizzare l’elenco dei nostri repository su Github. Si seleziona quel repository che ci interessa sul quale intendiamo lavorare e si procede con l’azione di Commit del Google Doc.

L’operazione è molto semplice e alla portata di tutti perché non richiede particolari conoscenze specifiche.

### 4.3.4 Il ‘Markup Panel’ e le relative funzioni

Cliccando su su «Show Markup Panel» viene visualizzato questo pannello.



che ci consente di inserire sul documento in Google Doc:

- Note colorate, personalizzabili nel titolo;
- Codice da illustrare in una pagina HTML;
- Tabella dell'indice dei contenuti (cioè il titolo delle pagine che compongono l'indice dei contenuti).

### 4.3.5 La costruzione dell'indice del documento

Dal Markup Panel di GGEditor è possibile costruire l'**indice** del documento da visionare sulle pagine HTML del progetto su Read the Docs.

Una volta inserita la maschera come di seguito rappresentato, basta editare il titolo delle altre pagine Google Doc che avete creato dentro la stessa directory di Google Drive. Sul file `.rst` che GGEditor crea verrà creato un indice che a sua volta verrà visualizzato su Read the Docs.

La procedura è di estrema facilità.

Dopo `:caption:` è possibile scrivere il nome che si vuole dare all'**indice**, o semplicemente scrivere "indice". Ogni volta che si crea un nuovo Google Doc da agganciare all'indice basta riportare il titolo del Google Doc in questa maschera.

```
.. toctree::
:maxdepth: 3
:caption: Indice

gdocs-rtd
tutorial
come-usarlo
lavoro-github
user-guide
licenza
```

la sintassi da editare per creare l'indice è la seguente

```
.. toctree::
:maxdepth: 3
:caption: Indice
gdocs-rtd
tutorial
come-usarlo
lavoro-github
lavoro-rtd
user-guide
hypothesis-partecipazione
pubblicare-su-docs-italia
sintassi-rst
tabelle
licenza
```

### 4.3.6 Inline Markups - Marcatori in linea

Some inline reStructuredText markups can be used directly in the document. The table below shows the usage example of these inline markups.

In Google Docs document	Rendered in HTML page
A `single back-quote`	A <i>single back-quote</i>
A ``double back-quote``	A double back-quote
A  replacement  markup	A <b> replacement </b> markup

That is converted from the source content in document:

Please noted that if you manually put a substitution markup, you got to provide correct replacement markup manually too. Otherwise, the sphinx parser will raise exception.

### 4.3.7 Inserimento di blocchi di codice

#### blocco di codice senza righe numerate

```
#!/usr/bin/env python

"""
Twisted moved the C{twisted} hierarchy to the C{src} hierarchy, but C{git}
doesn't know how to track moves of directories, only files. Therefore any
files added in branches after this move will be added into ./twisted/ and need
to be moved over into
"""

import os
from twisted.python.filepath import FilePath

here = FilePath(__file__).parent().parent()
fromPath = here.child("twisted")
toPath = here.child("src")

for fn in fromPath.walk():
    if fn.isfile():
        os.system("git mv {it} src/{it}"
                  .format(it="/" .join(fn.segmentsFrom(here))))

os.system('git clean -fd')

def outer(x):
def indent_start(x):
    go start start
    go start end

def end(y):
    go end start
    go end end
```

### blocco di codice con righe numerate

```
1 #!/usr/bin/env python
2
3 """
4 Twisted moved the C{twisted} hierarchy to the C{src} hierarchy, but C{git}
5 doesn't know how to track moves of directories, only files. Therefore any
6 files added in branches after this move will be added into ./twisted/ and need
7 to be moved over into
8 """
9
10 import os
11 from twisted.python.filepath import FilePath
12
13 here = FilePath(__file__).parent().parent()
14 fromPath = here.child("twisted")
15 toPath = here.child("src")
16
17 for fn in fromPath.walk():
18     if fn.isfile():
19         os.system("git mv {it} src/{it}"
20                 .format(it="/" .join(fn.segmentsFrom(here))))
21
22 os.system('git clean -fd')
23
24 def outer(x):
25     def indent_start(x):
26         go start start
27         go start end
28
29     def end(y):
30         go end start
31         go end end
```

#### 4.3.8 Visualizzazione di messaggi di errore: ‘undefined’ o messaggio ‘in lingua taiwanese’. Come risolverli.

Se durante la procedura di commit dal Google doc a Github viene visualizzato ripetutamente un messaggio di “**undefined**” o un **messaggio di error in lingua taiwanese**, che non permette di portare a termine il commit, l’azione da compiere è la seguente:

andare in “**Componenti aggiuntivi**”, poi “**GGeditor**” e quindi su “**Setting**” dove si trova un tasto rosso di “**Reset**” che cancella tutte le informazioni di collegamento agli account Github e ai relativi file nel repository. In effetti quello che avviene è una pulizia della cache, è come se si fosse appena installato il componente aggiuntivo GGeditor sul Google doc.

In vengono illustrati anche i messaggi che possono apparire nel caso descritto.

### 4.3.9 Inserire note colorate di vario tipo

Attraverso il tasto ‘Show markup panel’ è possibile inserire nel doc, e quindi nel progetto “read the docs” alcuni box per attirare l’attenzione del lettore su alcuni contenuti della pubblicazione. Eccoli di seguito elencati.

**Attenzione:** (content of Attention)

**Attenzione:** (content of Caution)

**Avvertimento:** (content of Warning)

**Pericolo:** (content of Danger)

**Errore:** (content of Error)

---

**Suggerimento:** (content of Hint)

---

---

**Importante:** (content of Important)

---

---

**Suggerimento:** (content of Tip)

---

---

**Nota:** (content of Note)

---

**Vedi anche:**

(content of See also)

---

**Change-me**

(content of Change-me)

---

### 4.3.10 Inserire in alcune pagine la possibilità di far lasciare commenti

In alcune pagine HTML del progetto Read the Docs, l'editore del documento da pubblicare online potrebbe avere l'esigenza di raccogliere commenti.

In questo caso, cioè solo quando questa esigenza dei commenti è relativa ad alcune pagine HTML del documento e non in tutte, viene in aiuto il servizio online di . Si procede innanzitutto creando un account in questo servizio e successivamente si crea un progetto, fornendo un URL a cui agganciare il servizio di Disqus. Lo stesso Disqus fornisce un codice HTML che bisogna copiare e incollare dentro un box HTML all'interno del Google doc nel quale si desidera far lasciare commenti.

Il codice Disqus di questa pagina ad esempio è il seguente ed è editato alla fine di questo Google doc che genera la pagina HTML sul progetto di Read the Docs:

```
<script id="dsq-count-scr" src="//guida-readthedocs.disqus.com/count.js" async></script>

<div id="disqus_thread"></div>
<script>

/**
 * RECOMMENDED CONFIGURATION VARIABLES: EDIT AND UNCOMMENT THE SECTION BELOW TO INSERT
 * DYNAMIC VALUES FROM YOUR PLATFORM OR CMS.
 * LEARN WHY DEFINING THESE VARIABLES IS IMPORTANT: https://disqus.com/admin/
 * universalcode/#configuration-variables*/
**/

var disqus_config = function () {
this.page.url = PAGE_URL; // Replace PAGE_URL with your page's canonical URL variable
this.page.identifier = PAGE_IDENTIFIER; // Replace PAGE_IDENTIFIER with your page's
 * unique identifier variable
};
*/
(function() { // DON'T EDIT BELOW THIS LINE
var d = document, s = d.createElement('script');
s.src = 'https://guida-readthedocs.disqus.com/embed.js';
s.setAttribute('data-timestamp', +new Date());
(d.head || d.body).appendChild(s);
})();
</script>
<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref_noscript
 * ">comments powered by Disqus.</a></noscript>
```

### 4.3.11 Altre (tante) utili funzioni di GGeditor



Inline Markups, Table, Image, Conversion.

e sono descritte molte funzioni che possono essere attivate con GGeditor, quale per esempio quella della **conversione del contenuto del nostro Google Doc in un file formato RST**, quindi con la sintassi tipica di questo linguaggio.

### 4.3.12 Conversione di testo da Google doc a file .RST per il download

Il componente aggiuntivo GGeditor permette anche la funzione di conversione del testo in linguaggio .RST (vedi per le funzioni complete di conversione). Praticamente è possibile, tramite una finestra dedicata, far convertire a GGeditor testo direttamente in linguaggio .RST. Si può convertire tutto il testo, una parte, o ad esempio una tabella. Ci sono delle regole di conversione già illustrate nella stessa finestra denominata "Conversion". Una volta convertito il testo appare un messaggio di avvenuta conversione ed è possibile effettuare il download del testo convertito in formato .RST oppure selezionarlo e copiarlo in un editor testuale per un ulteriore riuso.

### Conversion

```
.. _h666a272f165f5561d466c3f12192b2f:  
  
Le funzioni  
*****  
  
.. _h79285c4e7257194524363d5a1e617d44:  
  
Per i principianti dei file RST  
=====
```

## 4.4 Inserimento di tabelle nel documento

Creare tabelle da esporre in pagina HTML su Read the Docs, utilizzando il linguaggio rST è un argomento non molto semplice.

Se la tabella è elementare e piccola, ad esempio costituita da poche colonne, l'uso di un servizio online come può essere la soluzione più veloce. Terminato l'editing dentro le celle si procede con la selezione del tasto reStructuredText e si ottiene il testo già formattato per essere usato su un file formato rST.

Se le tabelle si devono editare nativamente con la sintassi rST, la cosa si fa più complicata, vedi questa spiegazione :  
editando questi caratteri:

```
+---+---+---+  
| a | b | c |  
+---+---+---+  
| e | f | g |  
+---+---+---+  
| h | i | l |  
+---+---+---+  
| m | n | o |  
+---+---+---+
```

si ha il seguente risultato

a	b	c
e	f	g
h	i	l
m	n	o

### 4.4.1 Con Ggeditor inseriamo tabelle nel Google doc

Con Ggeditor possiamo inserire tabelle dentro il nostro documento Google, quindi nel momento del “commit” la tabella sarà automaticamente trasformata in sintassi rST dentro il file rST di Github.

Al momento del “commit” Ggeditor ci dà due possibilità:

- Tables are converted to reStructuredText table markups.
- . This option would generate table with HTML <TABLE>. Useful for those who utilizes the readthedocs.org as a blog system.

---

**Suggerimento:** You can set background-color for header rows by assign CSS in the following file inside Github repo: `/docs/static/theme_overrides.css`.

For example:

```
.wy-table-responsive table th {  
    background-color: #f0f0f0;  
}
```

---

### 4.4.2 Creare la tabella su Google spreadsheet e poi importarla dentro Google doc

L'uso di di Drive per incorporare tabelle sulle pagine HTML può offrire una soluzione comoda, anche se didatticamente non rappresenta la soluzione ideale. Al fine di rendere l'uso di tabelle un'azione diffusa anche tra chi non ha consolidate esperienze nel campo della sintassi del linguaggio reStructuredText, la tabella del foglio Google rappresenta una via facile per l'esposizione su pagine HTML di Read the Docs.

Vediamo come.

Chiunque è in grado di creare una tabella su Google spreadsheet. Utilizziamo liberamente anche i colori per delimitare le celle e per evidenziare il contenuto delle celle o del testo. Prendiamo ad esempio questa tabella:

#### **Pubblica la tabella sul web**

La seconda azione da fare, dopo aver terminato l'editing nella tabella, è quella di pubblicare la tabella sul web. Le operazioni da compiere sono:

- seleziona in alto **“file”**
- selezione **“pubblica sul web”**
- clicca su **“link”** e seleziona **“pagina web”**
- seleziona e copia l'indirizzo url generato:

### Con GEditor Markup Panel si crea un box HTML

Si incorpora l'indirizzo URL sopra riportato nella pagina HTML, quindi si edita:

```
<iframe width=>100%<> height=>900px<> frameborder=>0<> src="indirizzo url"></iframe>
```

```
<iframe width="100%" height="900px" frameborder="0" src="https://docs.google.com/
↳spreadsheets/d/e/2PACX-1vRrShxVf6VZYXPeHR9e3NXsYZ_
↳x8nrE1gTuhqao4ERRm1XDYuXB07G4vqDkk4u96BfLRAjekwZPk3K/pubhtml?widget=true&
↳headers=false"></iframe>
```

dove **100%** rappresenta l'ampiezza (larghezza) della tabella, e **900px** rappresenta l'altezza della colonna in pixel che troverà posto nella pagina HTML di Read the Docs.

Il risultato sarà questo di seguito illustrato, cioè una tabella incorporata nella pagina HTML di Read the Docs.

L'unica cosa a cui porre attenzione è selezionare il limite destro della tabella al fine di farla rientrare interamente nella pagina HTML. Con 2-3 prove da fare si capirà quale è il limite massimo oltre il quale non andare.

E' una soluzione che permette di esporre tabelle avendo un file Google spreadsheet di origine che possiamo aggiornare quando vogliamo, e che si aggiornerà automaticamente sempre sulle pagine HTML di Read the Docs.

## 4.5 Inserimento di video, immagini e embedding nel documento

### 4.5.1 Inserimento di immagini

Il componente aggiuntivo GEditor permette all'utente di inserire immagini nello stesso documento. Lo stesso GEditor pensa a creare dei file immagine dentro la cartella `static` nel repository creato su Github.

Tuttavia inserendo immagini in questa modalità, è stato riscontrato che nella visione da dispositivi mobili, le immagini tendono a comprimersi in maniera casuale, risultando non di gradevole impatto visivo.

Per superare questa criticità la procedura da seguire per inserire immagini nel documento Google è la seguente:

- cliccare "componenti aggiuntivi",
- cliccare "GEditor"
- cliccare "Show Markup Panel"
- cliccare sull'ultimo componente elencato, cioè "Embed HTML (tag,js...)"

A questo punto dentro il box di colore bianco, andrà inserito il codice seguente

```

```

dove l'**url** è quello dove si trova caricata (online) l'immagine.

E' buona prassi caricare le immagini nella cartella `static` o in una cartella specificatamente dedicata, dentro il repository di Github, a raccogliere le immagini, e poi cliccando su ogni immagine, con tasto destro del mouse, cliccare su "copia l'indirizzo dell'immagine" e incollarlo nel box HTML dentro i doppi apici.

Se si ha esigenza di dare all'immagine una misura in larghezza, dopo l'url dentro i doppi apici si inserirà ad esempio `width=600` dove 600 è il numero dei pixel in altezza dell'immagine dentro la pagina HTML del progetto Read the Docs.

Inserendo immagini attraverso l'uso del box HTML le stesse verranno visualizzate nelle varie dimensioni dei display (pc desktop, tablet, smartphone) senza modifiche adattandosi automaticamente alle dimensioni.

## 4.5.2 Inserimento di video (embedding)

E' possibile inserire video all'interno dei documenti Google da far visualizzare poi nelle pagine HTML del progetto Read the Docs.

Nel caso di video di Youtube, l'azione da compiere è:

- cliccare "componenti aggiuntivi",
- cliccare GGeditor
- cliccare Show Markup Panel
- cliccare sull'ultimo componente elencato, cioè "Embed HTML (tag,js...)

A questo punto dentro il box HTML di colore bianco, andrà inserito il codice seguente

```
<iframe width="100%" height="500" src="https://www.youtube.com/embed/PUswAbvpE7c"
↪frameborder="0" allow="autoplay; encrypted-media" allowfullscreen></iframe>
```

dove nella parte `https://www.youtube.com/embed/PUswAbvpE7c` l'ultima parte dopo `/embed/` cioè `PUswAbvpE7c` è quella che identifica univocamente il singolo video che vogliamo inserire nel documento da pubblicare.

Nel caso di video di **Vimeo**, la stessa piattaforma fornisce un codice per effettuare la modalità di **embedding** (incorporamento). Basta inserire il codice fornito da Vimeo per ogni video nel box HTML dentro il Google doc.

## 4.5.3 Inserimento di mappe interattive (embedding)

Per ottenere, nel documento da pubblicare, una mappa interattiva come questa di seguito mostrata, è necessario inserire un box HTML (tramite lo strumento del "Show Markup Panel") all'interno del quale inserire il seguente codice:

```
<iframe width="100%" height="600px" frameBorder="0" allowfullscreen src="https://umap.
↪openstreetmap.fr/it/map/spazi-verdi-fruibili-a-palermo-italia_14577#12/38.1529/13.3673?
↪scaleControl=false&miniMap=false&scrollWheelZoom=false&zoomControl=true&
↪allowEdit=false&moreControl=true&searchControl=null&tilelayersControl=null&
↪embedControl=null&datalayersControl=true&onLoadPanel=caption&captionBar=false"></
↪iframe><br><a href="https://umap.openstreetmap.fr/es/map/spazi-verdi-fruibili-a-
↪palermo-italia_14577">Visualizza a schermo intero</a>
```

dove `height="600px"` indica l'altezza del riquadro che include la mappa, cioè quella mappa avrà un'altezza di 600 pixel e tale dato può essere variato in ragione delle esigenze del redattore del documento.

Ovviamente possono essere incorporate nel documento mappe provenienti da diverse piattaforme online (Google maps, uMap, Openstreetmap,...).

## 4.6 Il lavoro da fare su Github

### 4.6.1 1. Creare il nome del progetto

Creare il nome del progetto.

Scrivere nel file `README` la descrizione di cosa contiene quel progetto (un po di metadattazione del progetto). Come nel caso del .

Il nome del progetto sarà richiamato dal plugin GEditor una volta richiamato l'account Github ed editati user e password. Appariranno su un pannello i vari repository Github del nostro account e si sceglierà quello a cui inviare i Google Doc ai quali lavoreremo.

### 4.6.2 2. Creare un file “conf.py”

Creare un file dandogli il nome `conf.py` e all'interno incollare il seguente codice

```
1 # -*- coding: utf-8 -*-
2
3 from __future__ import unicode_literals
4 import sys, os
5 import re
6
7 from sphinx_rtd_theme import __version__ as theme_version
8 from sphinx_rtd_theme import __version_full__ as theme_version_full
9
10
11 on_rtd = os.environ.get('READTHEDOCS', None) == 'True'
12
13 sys.path.append(os.path.abspath(os.pardir))
14
15 __version__ = '1.0'
16
17 # -- General configuration -----
18
19 source_suffix = '.rst'
20 master_doc = 'index'
```

(continues on next page)

(continua dalla pagina precedente)

```

21 project = 'CHANGE-THIS'
22 copyright = '2016, CHANGE-THIS'
23
24 # The name of the Pygments (syntax highlighting) style to use.
25 pygments_style = 'sphinx'
26
27 extlinks = {}
28
29 # -- Options for HTML output -----
30
31 html_theme = 'sphinx_rtd_theme'
32
33 html_static_path = ['static']
34
35 def setup(app):
36     # overrides for wide tables in RTD theme
37     app.add_stylesheet('theme_overrides.css') # path relative to static
38
39     """
40     You might want to uncomment the "latex_documents = []" if you use CKJ characters in
41     ↪ your document.
42     Because the pdflatex raises exception when generate Latex documents with CKJ
43     ↪ characters.
44     """
45     #latex_documents = []
46
47     # inserire un logo in alto a sinistra (mettendo l'immagine nella cartella "static")
48     latex_logo = "static/immagine.jpg"
49     html_logo = "static/immagine.jpg"

```

**Importante:** Nella pagine del codice sostituire i seguenti parametri:

```
project = 'CHANGE-THIS'
```

al posto di CHANGE-THIS (dentro gli apici) editare il nome del titolo progetto che si vuole far comparire su Read the Docs;

```
copyright = '2016, CHANGE-THIS'
```

al posto di CHANGE-THIS (dentro gli apici) editare il tipo di licenza che si intende adottare per il rilascio della pubblicazione su Read the Docs.

### Creare il logo in alto a sinistra

Se si vuole far visualizzare un logo sulla parte in alto a sinistra, si procede in questo modo: alla fine del codice del file `conf.py`, bisogna scrivere:

```
latex_logo = "static/immagine.jpg"
html_logo = "static/immagine.jpg"
```

dove `immagine.jpg` sarà l'immagine da visualizzare in alto a sinistra (logo del nostro progetto), che metteremo dentro la cartella `static`.

### 4.6.3 3. Creare il file “theme\_overrides.css” e inserirlo dentro la cartella “static”

Il file `theme_overrides.css` serve per modificare la grafica base del file `conf.py`.

Serve anche per ottimizzare la visualizzazione delle tabelle ampie sulle pagine HTML di Read the Docs.

Si crea questo file dentro la directory `static`. Basta copiare il codice qui di seguito descritto in un file che chiameremo, appunto, `theme_overrides.css` dentro la cartella `static`.

```
1 .wy-table-responsive table td, .wy-table-responsive table th {
2     white-space: inherit;
3 }
4
5 .wy-table-responsive table th {
6     background-color: #f0f0f0;
7 }
8
9 .line-block, .docutils.footnote {
10     line-height: 24px;
11 }
12
13 .admonition {
14     margin-bottom: 20px;
15     line-height: 24px;
16 }
17
18 .admonition > *:not(:first-child){
19     /* draw a border around a admonition */
20     border-left: solid 1px #b59e9e;
21     border-right: solid 1px #b59e9e;
22     padding: 12px;
23     margin: -12px -12px -12px -12px;
24     margin-bottom: -12px !important;
25 }
26 .admonition > .last, .admonition- > .last{
27     /* draw a border around a admonition */
28     border-bottom: solid 1px #b59e9e !important;
29 }
30
31 /* adatta il nav-content a fullhd e si elimina lo spazio vuoto piccolo a destra */
32 .wy-nav-content {max-width: 1920px;}
```

(guarda ).

L'istruzione `.wy-nav-content {max-width: 1920px;}` consente di estendere per tutta la larghezza del monitor lo spazio in cui viene visualizzato il testo, dando una sensazione grafica gradevole all'intero documento.

#### 4.6.4 Come fare leggere un file in formato .MD a ReadtheDocs (in un progetto su Github)

()

Se voglio far leggere un file .MD a Read the Docs, oltre ai file .RST, quali impostazioni devo settare e dove li devo settare su Github?

Bisogna guardare questi due file `requirements.txt` e `conf.py` sul progetto ospitato da Github.

`requirements.txt`

`requirements.txt` è il file che contiene i requisiti dei moduli da installare. Vedi ad esempio: <https://github.com/opedatasicilia/tansignari/blob/master/requirements.txt>. Bisogna inserire queste istruzioni di seguito elencate nel file:

```
sphinx-rtd-theme
sphinx
recommonmark
markdown
sphinx-markdown-tables
```

–

`conf.py`

`conf.py` è il file di configurazione in linguaggio Python. In queste linee si imposta la configurazione che abilita il Markdown (.MD) su Read the Docs.

```
import recommonmark
from recommonmark.transform import AutoStructify

from recommonmark.parser import CommonMarkParser

source_parsers = {
    '.md': 'recommonmark.parser.CommonMarkParser',
}

source_suffix = ['.rst', '.md']

extensions = ['sphinx.ext.ifconfig', 'sphinx_markdown_tables']
```

Bisogna inserire queste righe di codice nel file `conf.py`.

#### 4.6.5 Una configurazione leggera ed efficace

Con le tre azioni di sopra si conclude tutto il lavoro da svolgere su Github, quindi questa soluzione si presta a chi non ha dimestichezza con il linguaggio RST.

Una configurazione del progetto Github molto leggera ma efficace in termini di risultati di pubblicazione di un documento sul design Read the Docs.

cirospat Update conf.py	
static	add licenza_1.png@2018-3-31 12:12:14
README.md	Update README.md
come-usarlo.rst	Google Docs reST Editor@2018-3-31 12:11:7
conf.py	<a href="#">Update conf.py</a>
index.rst	Google Docs reST Editor@2018-3-31 21:7:14
lavoro-github.rst	Google Docs reST Editor@2018-3-31 15:54:18
licenza.rst	Google Docs reST Editor@2018-3-31 20:52:2
tutorial.rst	Google Docs reST Editor@2018-3-30 15:7:0
user-guide.rst	Google Docs reST Editor@2018-3-30 22:30:4

Come si nota dall'elenco dei file che vengono generati dal plugin GGeditor direttamente nel repository Github abbiamo:

- un file `README.md` che è un file di descrizione del progetto, che provvediamo a editare noi su Github per far capire al lettore che cosa contiene il repository Github in questione. Questo file lo creiamo noi;
- il file `conf.py` che contiene il codice con indicazioni necessarie all'esposizione dei Google Docs sulla piattaforma di Read the Docs. . Basta creare un file nel repository Github, dargli il nome di `conf.py` e fare un copia e incolla dal paragrafo del tutorial di GGeditor. Questo file lo creiamo noi;
- una directory `static` che contiene soltanto immagini `.png` che sono le immagini che incolliamo nel Google Doc e che nell'azione del “**Commit**”, avviata dal plugin GGeditor, vengono generate automaticamente e inviate nella cartella `static`. Questa cartella `static` viene generata automaticamente dal plugin GGeditor;
- il file `theme_overrides.css` che creeremo dentro la directory `static`.
- i file `.rst` che sono i Google Doc convertiti automaticamente in file `.rst` dal plugin GGeditor e inviati nel repository Github. Questi file vengono generati dal componente aggiuntivo di Google doc, GGeditor;

Dalla descrizione di questi file si comprende come l'intero pacchetto su Github è molto semplice come tipologia di file. L'unico più complesso da capire è il contenuto del file `conf.py` e del file `theme_overrides.css` ma sono file che non dobbiamo nemmeno creare, perchè copiamo i contenuti dei file dal tutorial, andando a cambiare al suo interno solo il “nome” del documento da pubblicare e “il tipo di licenza” (questo solo nel file `conf.py`) e aggiungendo il nome del file immagine qualora volessimo creare un logo del progetto da far visualizzare in alto a sinistra sul progetto di Read the Docs.

#### 4.6.6 Inserire un logo in alto a sinistra nel design Read the Docs

Per inserire un immagine in alto a sinistra nel design di “Read the Docs”, per creare il logo del nostro progetto, basta andare nel file `conf.py` e alla fine inserire questo codice:

```
latex_logo = "static/immagine.png"
html_logo = "static/immagine.png"
```

avendo cura di caricare il file `immagine.png` nella cartella `static`.

**Attenzione:** Queste istruzioni non possono essere dati ai documenti da pubblicare in stile **Docs Italia**, ma solo ai documenti da pubblicare nello stile di base Read the Docs.

#### 4.6.7 Inserire la freccia “back to the top” nella pagina HTML



Al fine di permettere di risalire rapidamente in alto nella pagina HTML, torna comoda l'icona a forma di freccia sulla parte destra in basso della stessa pagina.

Di seguito la procedura per ottenere la freccia “back to the top” sulla pagina HTML.

Creare la cartella `_templates` e all'interno di essa creare il file `layout.html` e copiare il seguente codice:

```
1 <link href="{{ pathto("_static/theme_overrides.css", True) }}" rel="stylesheet" type=
  ↳ "text/css" />
2
3 <!-- css back top -->
4 <!--<link href="../_static/jqueryscriptptop.css" rel="stylesheet" type="text/css" />-->
5 <link href="{{ pathto("_static/jqueryscriptptop.css", True) }}" rel="stylesheet" type=
  ↳ "text/css" />
6 <!--<link href="../_static/backTop.css" rel="stylesheet" type="text/css" />-->
7 <link href="{{ pathto("_static/backTop.css", True) }}" rel="stylesheet" type="text/css" /
  ↳ >
8 <!-- Script -->
9 <!--<script type="text/javascript" src="../_static/jquery.min.js"></script> -->
10 <script type="text/javascript" src="{{ pathto("_static/jquery.min.js", True) }}"></
  ↳ script>
11 {% endblock %}
12
13
14 {% block footer %}
15     {{ super() }}
16     <!-- script Back To Top -->
17     <div class="footerc">
18         <a id='backTop'>Back To Top</a>
19     <!-- script toptoback automatico mobile/desktop -->
```

(continues on next page)

(continua dalla pagina precedente)

```
20 <!-- <script type="text/javascript" src="../../_static/jquery.backTop.min.js"></script> -->
21 <script type="text/javascript" src="{{ pathto("_static/jquery.backTop.min.js", True) }}">
    ↳</script>
22 <script>
23     $(document).ready( function() {
24         $('#backTop').backTop({
25             'position' : 400,
26             'speed' : 300,
27             'color' : 'green', <!-- lo sfondo freccia è di colore verde -->
28
29                                     });
30     });
31 </script>
32 </div>
33
34 {% endblock %}
```

Nella cartella `static` creare i file:

- 
- 
- 
- 

copiando il codice dai rispettivi file dei link di sopra.

Sempre dentro la cartella `static`, bisogna inserire un'immagine (con la freccia in alto) come questa contenuta .

E infine non dimenticare di inserire nel file `conf.py` alla fine delle righe, il seguente codice:

```
templates_path = ['_templates']
```

### 4.6.8 Inserire lateralmente icone per la condivisione delle pagine HTML di Read the Docs sui social network

Al fine di permettere la condivisione delle pagine del documento Read the Docs sui social network, una delle soluzioni che graficamente si adatta meglio alla grafica delle pagine RTD è fornita dalla piattaforma gratuita . Una volta creato l'account su **addthis** è possibile creare icone personalizzate (colore, dimensione) per la visualizzazione delle stesse sulle pagine html di RTD. Una volta creato il banner, sempre nella piattaforma addthis andare su "get the code" e copiare il codice che viene fornito. Tale codice è di questo tipo

```
<!-- Go to www.addthis.com/dashboard to customize your tools -->
<script type="text/javascript" src="//s7.addthis.com/js/300/addthis_widget.js#pubid=ra-
    ↳5b4c36adc4260026"></script>
```

e va incollato nella pagina `layout.html` (dentro la cartella `_templates`) prima di `{% endblock %}`. Tutto qui.

### 4.6.9 Aggiungere un'immagine dinamica in stile Word Cloud (WordArt)

Con è possibile creare immagini composte da parole che al passaggio del mouse vengono evidenziate.

Cosa fare per incorporare queste word cloud dentro una pagina HTML del progetto Read the Docs?

1. Attivare un account su e cominciare a creare un progetto. Si può importare l'elenco delle parole anche da una pagina web tramite l'URL.
2. Nel file `layout.html` del progetto Github inserire lo script: `<script src="//cdn.wordart.com/wordart.min.js" async defer></script>`.
3. Nella pagina di Google doc di interesse inserite il codice che vi dà dopo la creazione del vostro progetto. In ordine: Salvate il vostro progetto (save) - Andate su "share" quindi cliccate su "Embed on a webpage".
4. Copiate il codice, che dovrebbe essere di questo tipo `<div style="width: 400px; height: 400px;" data-wordart-src="//cdn.wordart.com/json/e9h2wyb41pzf" data-wordart-show-attribution></div>`, sulla pagina Google doc tramite la funzione HTML del pannello Markup di Ggeditor. Quindi fate il solito commit su Ggeditor che trasmetterà la nuova funzione a Github e così automaticamente vedrete sulle pagine HTML di Read the Docs la vostra nuova Word Cloud

Ecco la Word Cloud

### 4.6.10 Inserire in fondo alla pagina HTML del progetto Read the Docs lo spazio Disqus per i commenti

È possibile inserire in fondo alla pagina HTML del progetto Read the Docs uno spazio destinato ai commenti degli utenti, come avviene per tantissimi blog.

Uno dei servizi online gratuiti e molto diffusi è . È necessario creare un account su questo servizio e creare un progetto con lo stesso nome del progetto Read the Docs.

Per ogni progetto creato su Disqus verrà fornito il seguente codice da inserire nella pagina `footer.html`:

```
<p>
<script id="dsq-count-scr" src="//guida-readthedocs.disqus.com/count.js" async></script>
<div id="disqus_thread"></div>
<script>
/**
 * RECOMMENDED CONFIGURATION VARIABLES: EDIT AND UNCOMMENT THE SECTION BELOW TO INSERT
 * ↪ DYNAMIC VALUES FROM YOUR PLATFORM OR CMS.
 * LEARN WHY DEFINING THESE VARIABLES IS IMPORTANT: https://disqus.com/admin/
 * ↪ universalcode/#configuration-variables*/
/**

var disqus_config = function () {
this.page.url = PAGE_URL; // Replace PAGE_URL with your page's canonical URL variable
this.page.identifier = PAGE_IDENTIFIER; // Replace PAGE_IDENTIFIER with your page's
 * ↪ unique identifier variable
};
*/
```

(continues on next page)

(continua dalla pagina precedente)

```
(function() { // DON'T EDIT BELOW THIS LINE
var d = document, s = d.createElement('script');
s.src = 'https://guida-readthedocs.disqus.com/embed.js';
s.setAttribute('data-timestamp', +new Date());
(d.head || d.body).appendChild(s);
})();
</script>
<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref_noscript
↵">comments powered by Disqus.</a></noscript>
</p>
```

dove la parte guida-readthedocs deve essere sostituita con il nome del progetto specifico individuato su **Disqus** per il documento da pubblicare con lo stile “Read the Docs”.

Il codice sopra illustrato deve essere inserito nella pagina footer.html prima delle seguenti righe di codice:

```
{%- block extrafooter %} {% endblock %}
</footer>
```

Guarda direttamente al file su GitHub per comprendere meglio.

### 4.6.11 Inserire il popup eu-cookie nei progetti «read the docs»

(istruzioni a cura di Giovan Battista Vitrano)

Per aggiungere il popup **eu-cookie** per dare visibilità dei contenuti concernenti la privacy, basta aggiungere i tre file script di seguito elencati nella cartella static del progetto Github:

- 
- 
- 

Inserire nel file (nel blocco principale) il codice html:

```
<!--eu-cooki-lawt →
<!-- <script type="text/javascript" src="{{ pathto("_static/jquery-2.1.3.min.js", True) }}
↵}"></script> -->
<script type="text/javascript" src="{{ pathto("_static/jquery-eu-cookie-law-popup.js", ↵
↵True) }}"></script>
<link href="{{ pathto("_static/jquery-eu-cookie-law-popup.css", True) }}" rel="stylesheet
↵" type="text/css" />
```

Sempre nel file layout.html inserire il seguente codice:

```
<div class="eupopup eupopup-top "></div>
```

Guarda il per il corretto inserimento del codice.

Per modificare il testo del popup, apri il file `jquery-eu-cookie-law-popup.js` con notepad++, o anche con il semplice notepad, e cerca il blocco **PARAMETERS (MODIFY THIS PART)**, e li modifichi url della pagina **privacy** ed il testo:

```
// PARAMETERS (MODIFY THIS PART) //////////////////////////////////////
_self.params = {
cookiePolicyUrl : 'https://cirospat.readthedocs.io/it/latest/privacy.html',
popupPosition : 'top',
colorStyle : 'default',
compactStyle : false,
popupTitle : 'Questo sito web utilizza i cookie, anche di terze parti, per migliorare la
↳vostra esperienza di navigazione web.',
popupText : 'Chiudendo questo banner, scorrendo questa pagina o cliccando su qualunque
↳suo elemento acconsenti all&rsquo;uso dei cookie. Per maggiori informazioni o per
↳negare il consenso a tutti o ad alcuni cookie, consulta l&rsquo;informativa.',
buttonContinueTitle : 'Chiudi! ',
buttonLearnmoreTitle : 'Leggi l&rsquo;informativa',
buttonLearnmoreOpenInNewWindow : false,
agreementExpiresInDays : 30,
autoAcceptCookiePolicy : false,
htmlMarkup : null
};
```

**Ricordati (!)** che devi aggiungere nel tuo progetto “Read the Docs” la pagina dell”. Ovviamente questa pagina HTML sarà del testo preventivamente editato su un Google doc.

#### 4.6.12 Cambiare il colore di sfondo del rettangolo in alto a sinistra

Cambiare colore sul rettangolo superiore in alto a sinistra, dove è situato il nome del progetto, è possibile. Qui di seguito si riporta il codice da inserire sul file `theme_overrides.css` che si trova dentro la cartella `static`:

```
}

.wy-side-nav-search {
    background-color: #525e99;
}
```

il codice “#525e99” usato in questo caso (il colore del rettangolo in alto a sinistra del tutorial che state leggendo) corrisponde alla tonalità cromatica verificabile a questo link:. Ovviamente cambiando codice numerico (con #iniziale) è possibile generare altre tonalità da applicare al caso specifico.

**Attenzione:** Queste istruzioni non possono essere dati ai documenti da pubblicare in stile **Docs Italia**, ma solo ai documenti da pubblicare nello stile di base Read the Docs.

### 4.6.13 Cambiare il colore dei titoli dei capitoli, paragrafi, sottoparagrafi, ecc.

Come prima, è anche possibile cambiare il colore dei titoli dei capitoli, paragrafi, sottoparagrafi, ecc. Sempre sul file `theme_overrides.css` si riporta il seguente codice:

```
}  
  
h1, h2, h3 {  
    color: #176a90 !important;  
}
```

il codice “” può essere cambiato con i codici di tantissimi altri colori disponibili.

**Attenzione:** Queste istruzioni non possono essere dati ai documenti da pubblicare in stile **Docs Italia**, ma solo ai documenti da pubblicare nello stile di base Read the Docs.

### 4.6.14 Inserire una barra di scroll orizzontale in alto nella pagina

Al fine di far visualizzare al visitatore della pagina a che livello (sull’intera pagina) è arrivato nella lettura, torna utile inserire un piccolo sottile scroll orizzontale da posizionare in alto.

Bisogna fare due cose. Prima cosa: incollare nel file `layout.html` il seguente codice, prima della riga in cui si trova `{% endblock %}`:

```
<!-- Reading Progress Bar on Scroll -->  
<!-- vedere questo repo per progress bar: https://github.com/mburakerman/prognroll -->  
<script type="text/javascript" src="{{ pathto("_static/prognroll.js", True) }}"></script>  
<script>  
    $(function() {  
        $("body").prognroll({  
            height: 4, //Progress bar height in pixels  
            color: "#3337c4", //Progress bar background color  
        });  
    });  
</script>
```

dove

- **‘height’** è l’altezza della barra dell scroll
- **‘color’** è il codice del colore che vogliamo assegnare alla barra dello scroll. A questo è possibile scegliere i codici di una vasta gamma di colori da utilizzare.

Seconda cosa da fare: creare un file Java Script `prognroll.js` nella cartella `static` dove inserire .

#### 4.6.15 Uno schema tipo di progetto Github che raccoglie tutte le funzioni illustrate in questa pagina del tutorial

A questa pagina di Github si trova uno “**schema tipo**” di repository la cui restituzione come progetto Read the Docs è disponibile qui: .

Lo schema tipo Github può essere clonato per creare un altro progetto Github che abbia le stesse impostazioni, che graficamente sono visibili nel relativo .

Quindi la funzione dello schema tipo Github è quella di facilitare tutte le procedure di editing del codice, non dovendo pensare a crearlo da zero, e dando la possibilità all’utente di cambiare le personalizzazioni (titolo del progetto e versione della licenza nel file `conf.py`, colore testo dei capitoli/paragrafi, colore del riquadro in alto a sinistra e altre impostazioni nel file `theme_override.css` dentro la cartella `static`) e di concentrarsi maggiormente sui contenuti (testo, immagini, video,..) della pubblicazione che saranno editati direttamente dentro i Google doc.

### 4.7 Il lavoro da fare su Read the Docs

Questa rappresenta la fase finale del lavoro ed è molto semplice come operazioni da effettuare. Una volta completato il lavoro di compilazione su Github, bisogna andare su e (dopo aver creato il relativo account) importare il progetto, già creato, da Github.

Nella finestra, su “URL del Deposito Codice”, bisogna scrivere l’URL del progetto che avete creato su Github, e quindi scegliere il nome del progetto, ad esempio:

“**linee guida open data comune vattelapesca**”

e lasciare “Tipo del Deposito Codice” selezionato su “Git”.

A quel punto verrà messo in collegamento il vostro progetto di **Github** con il progetto su **Red the Docs**.

Una primissima azione da compiere è andare su “Amministrazione” e settare la lingua italiana. Questo consentirà a Read the Docs di aggiungere al titolo del vostro progetto la desinenza `/it/latest`. Il documento è in italiano quindi prende la desinenza `/it`. Questa impostazione permetterà alle note colorate che avete creato su Google doc di avere un titolo italiano (“Nota” al posto di “Note”, “Avvertimento” al posto di “Warning”, “Attenzione” al posto di “Attention” ecc.).

Esempio: `come-creare-guida.readthedocs.io/it/latest`

A questo punto il progetto di Github è compilato su Read the Docs.

Considerato che avevamo scelto come titolo del nostro progetto su Read the Docs: “**linee guida open data comune vattelapesca**”, l’URL compilato da Read the Docs per vedere il nostro progetto sarà:

`linee-guida-open-data-comune-vattelapesca.readthedocs.io`

### 4.7.1 Messaggi di ‘passing’ e ‘failing’ sul pannello di controllo di Read the Docs

---

#### Avviso di passing

##### Procedura andata a buon fine: «passing»

Se non ci sono errori commessi durante le procedure spiegate fino ad ora, tutto andrà a buon fine, e Read the Docs darà il messaggio in colore verde di «**passing**» al nostro progetto, significa che il nostro progetto è - quindi - online. La compilazione (build) su Read the Docs avviene con successo e ogni modifica che effettuiamo sul file del Google doc, viene commissionato a Github e compilato in tempo reale su Read the Docs, apparendo immediatamente sulle pagine HTML. La “build” su Read the Docs viene eseguita correttamente.

---

#### Avviso di failing

##### Procedura con Errore: «failing»

Se sono stati commessi errori nella procedura finora illustrata, Read the Docs alla sezione “i miei progetti”, darà un messaggio in colore rosso di «**failed**». In questo caso c’è qualche problema da qualche parte e bisogna ripercorrere tutti i passaggi fatti da quando si è iniziato a lavorare a partire dai file su Google doc fino a quanto eseguito sul sito di Read the Docs. La compilazione su Read the Docs ha incontrato qualche problema, e quando si presenta questo caso la prima cosa da fare è andare nel file `conf.py` - dentro il repository del progetto su Github - e verificare le istruzioni date dentro questo file. Generalmente se si presenta un problema nella compilazione di Read the Docs, il problema sta dentro questo file. Una volta individuato e risolto il problema, Read the Docs comincerà automaticamente a compilare le istruzioni del file `conf.py` di Github e darà il bollino verde di «passed» (cioè la compilazione è effettuata con successo).

---

Abbiamo completato tutte le procedure e ci possiamo godere il nostro documento nella nuova modalità di pubblicazione e visualizzazione con lo stile Read the Docs o con il design Docs Italia.

[Questa pagina è del tutorial “” (a cura di Pablo Persico, Andrea Borruso e Ciro Spataro) ed ulteriormente arricchita].

### 4.7.2 Messaggio Read the Docs di “build fails”: [cannot import name “PackageFinder” from “pip.\_internal.index”]

Può capitare che durante la procedura di compilazione del progetto su RTD appaia un messaggio:

*“cannot import name “PackageFinder” from “pip.\_internal.index”.*

---

#### La causa

Currently all builds are failing because the automatic upgrade (since ) to pip 20.0 was buggy (see ). There’s now a 20.0.1 release which seems to have fixed the problem for others ... but how can I force my readthedocs to also upgrade to the .1 version?

---

#### La soluzione ():

Sometimes it happen that your Builds start failing because the build environment where the documentation is created is stale or broken. This could happen for a couple of different reasons like pip not upgrading a package properly or a corrupted cached Python package.

In any of these cases (and many others), the solution could be just wiping out the existing build environment files and allow Read the Docs to create a new fresh one.

Follow these steps to wipe the build environment:

- Go to Versions
- Click on the Edit button of the version you want to wipe on the right side of the page
- Go to the bottom of the page and click the wipe link, next to the “Save” button

Note: By wiping the documentation build environment, all the `rst`, `md`, and `code` files associated with it will be removed but not the documentation already built (HTML and PDF files). Your documentation will still be online after wiping the build environment.

Now you can re-build the version with a fresh build environment!

### 4.7.3 Web Analytics - inserire il codice su

E' possibile agganciare strumenti di web analytics ai progetti online di Read the docs. Se si usa, ad esempio Google Analytics, una volta creato il progetto specifico su , si ottenuto il codice. Il codice va inserito nel progetto specifico nel pannello di Amministrazione di read the docs, seguendo questo percorso:

Amministrazione / Impostazioni avanzate, e andando in fondo alla pagina fino alla voce Codice Analytics, quindi cliccare il testo 'salva'.

---

## 4.8 User-guide (gli step operativi)

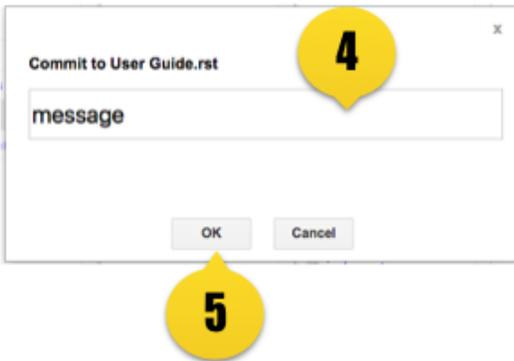
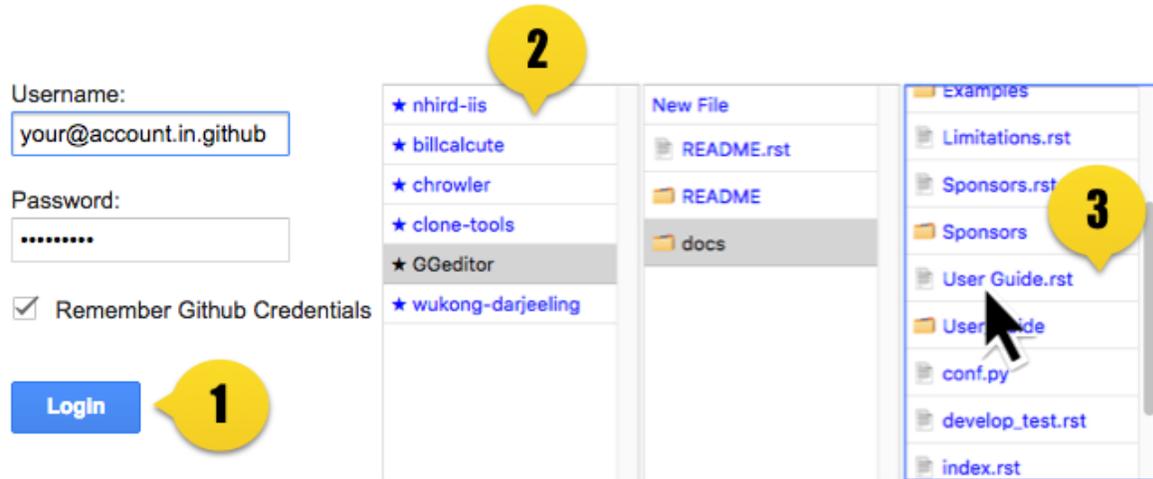
### 4.8.1 L'importanza della formattazione corretta del testo su Google doc

La formattazione corretta nel testo del Google doc è la cosa più importante da seguire. Appare molto importante rispettare il “titolo”, il “sottotitolo”, l’”intestazione 1”, l’”intestazione 2”, l’”intestazione 3”, e così via. Il resto del testo deve essere formattato come “testo normale”.

Se non rispettate le regole (facili) della formattazione, il testo in HTML nel progetto Read the Docs ne risentirà, non permettendo una facile lettura.

## 4.8.2 Consegnare i file da Google Doc a Github

Una volta effettuato l'accesso a Github (dall'interno di Google Doc) e terminata la compilazione del testo su Google Doc si passa all'operazione di commissionamento del file a Github. La finestra di GGeditor guida alla creazione del file di Google Doc dentro il progetto di Github. La procedura guidata dalla finestra di GGeditor è abbastanza intuitiva.



1. Logging your GitHub account
2. Navigate to repository
3. Click on the file to bind
4. Fill the committing message
5. Click "OK" to commit

Una volta cliccato OK per il commit su Github, verrà creato un file RST sulla cartella «doc» dentro il progetto di Github.

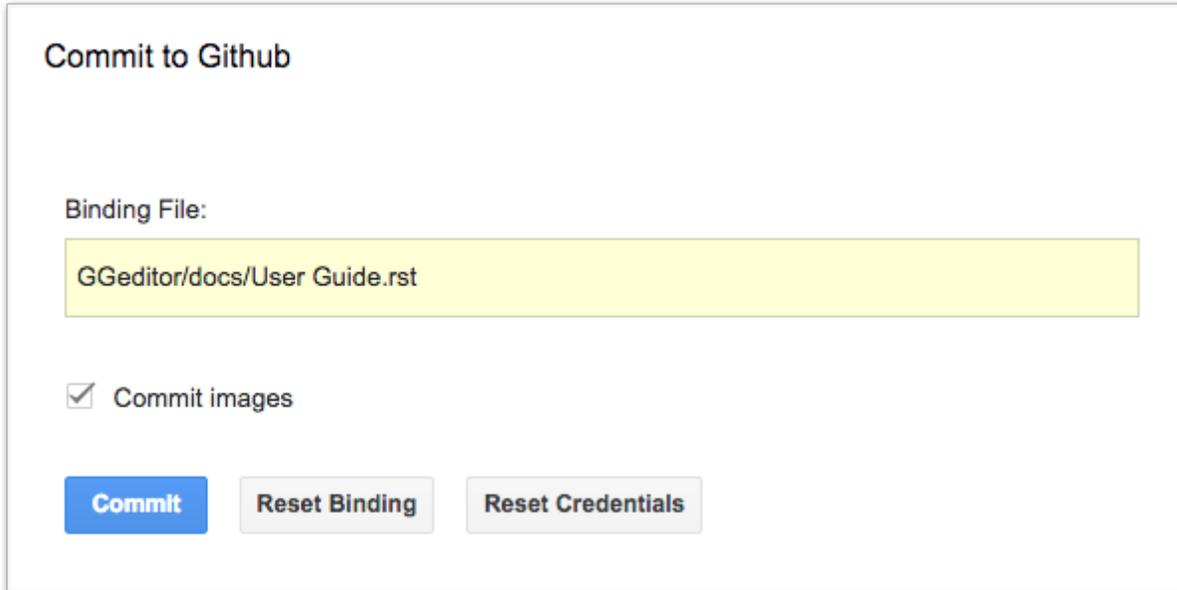
**Attenzione:** È importante sapere che prima di effettuare questa operazione illustrata nella figura di sopra, è necessario creare il nome del progetto su Github, in maniera tale che quando seguiamo le operazioni della figura (dalla 1 alla 5) indichiamo a Github il nome del progetto nel quale GGeditor deve andare a creare il file che prende lo stesso nome del titolo del Google Doc, in più GGeditor provvede ad aggiungere il suffisso «.rst» dentro la directory madre di Github.

If you want to commit to a new file. Please

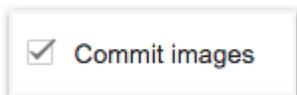
1. Navigate to the folder where the new file would be
2. Click on the "New File" item
3. Give the file name to create. The name will be suffix with ".rst" automatically.

## Committing

L'azione di “committing” rappresenta il trasferimento del contenuto del Google doc al file in formato `.rst` dentro il repository di Github. In questo trasferimento il contenuto del Google doc viene trasformato automaticamente in linguaggio `.rst`. Questo è il lavoro più prezioso che svolge il componente aggiuntivo GGeditor!



[la finestra che ci indica il Google Doc che dobbiamo inviare in una cartella («.doc») del progetto su Github]



Se solo il testo è stato modificato nel Doc, puoi evitare di flaggare il tasto “Commit images” per escludere il commit a Github delle immagini, questo porterà a una riduzione dei tempi dell’effettuazione del processo di commit.

**Nota:** GGeditor posizionerà i file immagini presenti nel Google doc sulla cartella `static` del progetto su Github. Se nel Google doc vengono modificate immagini o anche vengono cancellate o sostituite, ogni volta bisogna cliccare sul componente aggiuntivo GGeditor e quindi su «commit images».

**Attenzione:** Se avete molte immagini in un Doc, il che significa molte immagini da caricare su Github tramite GGeditor, potrebbe succedere di incontrare situazioni di immagini rotte nelle corrispondenti pagine HTML di Read the Docs. Questo perché il processo di rigenerazione delle pagine HTML è ancora in corso, non dovete -quindi- preoccuparvi. In questo caso dovete aspettare ancora un po che finisca il processo di costruzione delle stesse pagine su Read the Docs. Oppure se le immagini non compaiono ancora nelle pagine HTML, un ultima cosa da fare è andare nella pagina «**Administration**» del progetto su Read the Docs e cliccare su «**Saving**».

Comunque se vedi immagini vecchie e non rispondenti all’ultima versione del Doc, elimina la cache del browser o controlla sul repository di Github la corrispondenza delle foto del Google doc con quelle nella cartella `static` su Github.

### Vedi anche:

In merito all’inserimento delle immagini nel Google doc, vedere anche il paragrafo “”.

### 4.8.3 Limitazioni

da Google Doc sui file reST:

- Comments. This is not supported by the reST<sup>1</sup>.
- Drawing objects. Because there is no API to get it as an image.
- List styles. The list style is defined by the CSS settings in the html page.
- Math equations. Because this is no API to get it as an image.
- Multi-columns. This is not supported by the reST.
- Page break. This is not able to apply to a html page.
- Page header and page footer. This is not supported by the reST.
- Page numbering. This is not able to apply to a html page.
- Internal link to heading does not work. Currently there is no API to identify the target heading element. Please use “Bookmark“ instead.
- Bold and italic styles in footnote content does not exposed by Doc’s API. Which means bold and italic text is rendered as normal text in footnote content.

## 4.9 Strumenti per i più esperti

### 4.9.1 Conversione

Per la **conversione** del testo da Google Doc al formato RST (e anche previsto il download del file RST) si fa riferimento a .

### 4.9.2 Api Docs

Per **API document for a Python module** si fa riferimento a .

### 4.9.3 Backend

Documentazione specificata nel . Si fa .

---

<sup>1</sup> per i commenti usa il servizio di hypothes.is illustrato in [questa pagina de tutorial](#)

## 4.10 Hypothes.is per partecipare ad un documento su Read the Docs

Al fine di permettere la partecipazione di utenti che intendono commentare un documento pubblicato su Read the Docs viene in aiuto la piattaforma .

Di seguito vengono esposte dettagliatamente le procedure da seguire per integrare hypothes.is su Read the Docs.

A

Aggiungere `templates_path = ['_templates']` al file `conf.py`.

B

Creare nel repo una cartella `_templates`, e all'interno della cartella creare un file `layout.html` ed in questo file inserire il seguente codice:

```
{% extends "!layout.html" %}
{% block extrahead %}
{{ super() }}
<script src="https://hypothes.is/embed.js" async></script>
{% endblock %}
```

Tutto qui.

La procedura è stata definita da al quale va un ringraziamento per questa importante integrazione su Read the Docs. Andrea ha sperimentato per la prima volta l'applicazione di questa procedura sul progetto del .

### 4.10.1 Le annotazioni su hypothes.is disponibili in tempo reale in un feed RSS o in un formato Json

C'è un feed RSS per le note di ogni pagina HTML. Ad esempio:

`url-di-readthedocs / pagina-specifica-del-documento-readthedocs.html`

**Nota:** Guarda, ad esempio, il feed RSS delle annotazioni postate dagli utenti come commenti ad una pagina del Libro bianco dell'innovazione della PA 2018:

Esempio di visualizzazione del feed rss dei commenti ad una pagina (html) del Libro bianco innovazione PA 2018

```
<?xml version="1.0"?>
<rss version="2.0"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel>
    <title>Hypothesis Stream</title>
    <link>https://hypothes.is/stream</link>
    <atom:link href="https://hypothes.is/stream.rss" rel="self" type="application/rss+xml" />
    <description>The Web. Annotated</description>
    <pubDate>Sat, 15 Sep 2018 10:15:42 -0000</pubDate>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
```

(continues on next page)

```

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.
↳ForumPA documentazione</title>
  <description>gli acquisti vanno anche agevolati soprattutto quelli del
↳funzionamento delle segreterie che sono speso messi da parte costringendo il personale,
↳ad acquistare con il proprio stipendio la cancelleria mancante, evitando gli sprechi,
↳con possibilità di fax telefoni fotocopiatrici scanner o altro a disposizione del
↳dipendente che lavora nella segreteria oltre agli arredi base </description>
  <pubDate>Sat, 15 Sep 2018 10:15:42 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:TXlfhLjQEeitoSumpIwxIw</guid>
  <link>https://hypothes.is/a/TXlfhLjQEeitoSumpIwxIw</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.
↳ForumPA documentazione</title>
  <description>Si ricorda soprattutto di tenere conto di tutto il cartaceo esami,
↳lastre, tac nelle aziende sanitarie ospedaliere che anche dopo anni potrebbero tornare,
↳utili al paziente comprendendo anche il privato o convenzionato che abbiano una
↳piattaforma comune e che il tutto non sia trasportato lontano dal luogo di residenza e
↳di cura ma che sia tracciabile anche a distanza di anni </description>
  <pubDate>Sat, 15 Sep 2018 10:12:11 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:z97SPrjPEeiHFMTlTtD4UA</guid>
  <link>https://hypothes.is/a/z97SPrjPEeiHFMTlTtD4UA</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;modello organizzativo unico e diffuso&lt;/
↳blockquote&gt;che sia funzionante e che tenga conto dei lavoratori soprattutto a
↳livello umano </description>
  <pubDate>Sat, 15 Sep 2018 10:09:36 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:c1nP-rjPEeiGNROjgbIDzw</guid>
  <link>https://hypothes.is/a/c1nP-rjPEeiGNROjgbIDzw</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.
↳ForumPA documentazione</title>
  <description>nel rispetto e con rispetto dei lavoratori </description>
  <pubDate>Sat, 15 Sep 2018 10:08:27 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:Sh8NirjPEeiQB2N0A8sSng</guid>
  <link>https://hypothes.is/a/Sh8NirjPEeiQB2N0A8sSng</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

```

(continues on next page)

(continua dalla pagina precedente)

```

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;mancanza di una cultura della trasparenza &lt;/
↳blockquote&gt;la mancanza della cultura della trasparenza nasce dall&#39;incapacità_
↳organizzativa di diffondere informazioni, strategie, obiettivi.. si raggiungono gli_
↳obiettivi incuranti del personale e di come è stato coinvolto di solito ultimamente_
↳negativo in tutti i sensi (controlli personali, solo a determinate persone, verifica_
↳continua degli orari ed assillate soprattutto in strutture con pochissimo personale e_
↳mai aiutato con delle supplenze in caso di malattia, ferie,ecc. </description>
  <pubDate>Sat, 15 Sep 2018 10:03:14 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:j45mKLjOEeiFmJt0S9uMLg</guid>
  <link>https://hypothes.is/a/j45mKLjOEeiFmJt0S9uMLg</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;nuove tecnologie&lt;/blockquote&gt;tecnologie_
↳valide, facilmente utilizzabili da tutta Italia con pc, stampanti, ecc che dialogano e_
↳con un costo abbordabile e senza costi aggiuntivi per eventuali modifiche dell&#39;
↳utilizzo</description>
  <pubDate>Sat, 15 Sep 2018 09:57:24 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:vyFVQLjNEeizAk_yvufcTQ</guid>
  <link>https://hypothes.is/a/vyFVQLjNEeizAk_yvufcTQ</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;telelavoro o forma di conciliazione&lt;/blockquote&
↳gt;basta che la forma di conciliazione non si un ennesimo atto di mobbing, bossing,
↳pressing ecc. nei confronti dei lavoratori, potrebbero essere esclusi i dirigenti_
↳amministrativi e contabili, tecnici sanitari e di radiologia, ma non il normale_
↳personale amministrativo contabile che necessita di forme di conciliazione soprattutto_
↳dal compimento del 55° anno di età e cioè verso la pensione </description>
  <pubDate>Sat, 15 Sep 2018 09:55:45 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:g7h1sLjNEeiuQPM57QMZlg</guid>
  <link>https://hypothes.is/a/g7h1sLjNEeiuQPM57QMZlg</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;smart working (o Lavoro Agile) &lt;/blockquote&gt;
↳Lo smart working non dovrebbe essere adottato dai dirigenti soprattutto amministrativi_
↳e contabili per cui è necessaria la quotidiana presenza, ma solo dai dipendenti che_
↳abbiano una reale necessità dovuta alla salute e alla gestione della famiglia presente_
↳e passata e che non abbiamo parenti ed affini che si prestino in sostituzione alle_

```

(continues on next page)

```

↳difficoltà</description>
  <pubDate>Sat, 15 Sep 2018 09:49:55 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:szanDrjMEeiTZRvPWGYxjw</guid>
  <link>https://hypothes.is/a/szanDrjMEeiTZRvPWGYxjw</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.↳
↳ForumPA documentazione</title>
  <description>sempre con una visione di coinvolgimento umana, sociale e non di↳
↳esclusione, ma di inclusione anche dei dipendenti deboli o che hanno subito troppi↳
↳capovolgimenti di mansioni e ufficio nella loro non riconosciuta carriera </↳
↳description>
  <pubDate>Sat, 15 Sep 2018 09:46:17 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:MRhl0rjMEeiuPwOWtMJ9ug</guid>
  <link>https://hypothes.is/a/MRhl0rjMEeiuPwOWtMJ9ug</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.↳
↳ForumPA documentazione</title>
  <description>importanti che i costi delle tecnologie non siano esosi ed a caspito.↳
↳della comunità e della società ma che siano controllati per essere efficienti,↳
↳efficaci, semplici da utilizzare e con costi abbordabili </description>
  <pubDate>Sat, 15 Sep 2018 09:42:51 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:tssCvLjLEeiRDX_TrFg4Mw</guid>
  <link>https://hypothes.is/a/tssCvLjLEeiRDX_TrFg4Mw</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018.↳
↳ForumPA documentazione</title>
  <description>ottimo progetto, quello di Barcellona, assieme a CONSUL di Madrid.↳
↳Anche in Italia ci difendiamo bene - ci sono diverse piattaforme simili - ma sono↳
↳tutte iniziative civiche e non comunali (salvo per alcuni aspetti Bologna), come in↳
↳Spagna, in grado di dare sostenibilità e solidità a questi progetti.

Ciò che conta e distingue le piattaforme, dal mio punto di vista, è tuttavia la finalità↳
↳per cui sono messe online: segnalazioni, contributi...o processi decisionali? Oltre al↳
↳bilancio partecipativo - che permette ai cittadini la sperimentazione di un percorso↳
↳(e di una responsabilità) decisionale attraverso un budget - è possibile anche↳
↳prevedere dei processi di &#34;obbligo politico&#34; (o di pressione sociale)↳
↳attraverso cui le amministrazioni si impegnano non tanto a realizzare quanto chiesto↳
↳ma almeno affrontare un tema prioritario - rispondendo a quesiti o organizzando degli↳
↳incontri pubblici - che altrimenti rimarrebbe nel cassetto perché magari troppo↳
↳scomodo. </description>
  <pubDate>Sat, 15 Sep 2018 09:21:54 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:yYPunrjIEeif8AcEL2kapg</guid>
  <link>https://hypothes.is/a/yYPunrjIEeif8AcEL2kapg</link>

```

(continues on next page)

(continua dalla pagina precedente)

```

    <dc:creator><![CDATA[ste.sto]]></dc:creator>
  </item>

  <item>
    <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
    <description>&lt;blockquote&gt;Occorre avviare sperimentazioni serie e_
↳verificabili di auditing civico in diverse tipologie di enti, attraverso un_
↳investimento importante sia di risorse, sia di relazioni con i soggetti della_
↳cittadinanza organizzata. Occorre inoltre dare evidenza dei risultati delle_
↳sperimentazioni e discuterli con la dirigenza apicale degli enti.&lt;/blockquote&gt;
↳Auditing civico non può funzionare senza una buona e funzionante organizzazione che_
↳non tenga conto solo degli obiettivi da raggiungere ma che crei un buon ambiente di_
↳lavoro per il lavoratore e per la cittadinanza se raggiungi lo stesso obiettivo con_
↳due persone mi pare che la situazione sia squilibrata dove magari un altro obiettivo è_
↳stato raggiunto con 100 persone e magari con l#39;assunzione di personale giovane_
↳dirottato a favore di personaggi che non sanno gestire senza il loro apporto </
↳description>
    <pubDate>Sat, 15 Sep 2018 08:59:04 -0000</pubDate>
    <guid isPermaLink="false">tag:hypothes.is,2015-09:mKyilrjFEeiGK7fYjXIHFw</guid>
    <link>https://hypothes.is/a/mKyilrjFEeiGK7fYjXIHFw</link>
    <dc:creator><![CDATA[LyLilly]]></dc:creator>
  </item>

  <item>
    <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
    <description>&lt;blockquote&gt;un codice deontologico della professione. &lt;/
↳blockquote&gt;codice deontologico della professione che tenga conto anche del lato_
↳umano del dipendente </description>
    <pubDate>Sat, 15 Sep 2018 08:48:40 -0000</pubDate>
    <guid isPermaLink="false">tag:hypothes.is,2015-09:JI_NbLjEEeiHBzfC_7fkTg</guid>
    <link>https://hypothes.is/a/JI_NbLjEEeiHBzfC_7fkTg</link>
    <dc:creator><![CDATA[LyLilly]]></dc:creator>
  </item>

  <item>
    <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_
↳ForumPA documentazione</title>
    <description>&lt;blockquote&gt;compiti impossibili&lt;/blockquote&gt;compiti_
↳impossibili sempre diversi dalla segretaria, alla bibliotecaria, alla tecnica, alla &
↳#34;badante&#34; di soggetti anziani, ecc. ed inoltre, continui trasferimenti interni_
↳alla struttura per chiusure di uffici, o altro soprattutto per le persone over 50 anni_
↳sono deleteri a livello professionale ed umano</description>
    <pubDate>Sat, 15 Sep 2018 08:47:31 -0000</pubDate>
    <guid isPermaLink="false">tag:hypothes.is,2015-09:-209PrjDEeiP_ish9r9BBw</guid>
    <link>https://hypothes.is/a/-209PrjDEeiP_ish9r9BBw</link>
    <dc:creator><![CDATA[LyLilly]]></dc:creator>
  </item>

  <item>
    <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018_

```

(continues on next page)

```

↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt; i modi dell'attuazione.&lt;/blockquote&gt;I modi
↳di attuazione non devono essere fugaci ed invisibili, perchè la parte migliore viene
↳sempre offerta ad una parte di persone e gli altri sono allo scuro di tutto, inoltre,
↳deve sempre contenere una parte umana a favore del lavoratore che ha problemi di
↳famiglia o che ha avuti grazie </description>
  <pubDate>Sat, 15 Sep 2018 08:45:07 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:pdQtGrjDEeif61vIR4APdA</guid>
  <link>https://hypothes.is/a/pdQtGrjDEeif61vIR4APdA</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;coaching.&lt;/blockquote&gt;il coaching funziona
↳se c'è una base umana e non solo lavorativa </description>
  <pubDate>Sat, 15 Sep 2018 08:41:27 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:Ip4YtrjDEeitl5sQx-ID9g</guid>
  <link>https://hypothes.is/a/Ip4YtrjDEeitl5sQx-ID9g</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;retribuzioni&lt;/blockquote&gt;aumento delle
↳retribuzioni a chi non effettua visite in extramoenia oltre a 2000 euro in campo
↳medico, riconoscimento dei diritti ed esclusione dalle notti dei medici over60 anni
</description>
  <pubDate>Sat, 15 Sep 2018 08:40:14 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:9vPwZLjCEeiTYT9FhEe3wQ</guid>
  <link>https://hypothes.is/a/9vPwZLjCEeiTYT9FhEe3wQ</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018
↳ForumPA documentazione</title>
  <description>&lt;blockquote&gt;comunità educanti&lt;/blockquote&gt;Le comunità
↳educanti devo essere a loro volta umane e contrarie al bullismo, semimobbing, mobbing,
↳bossing , violenze psicologiche utilizzate per far carriera sia a livello di
↳personale non docente/dirigente che a livello di semplici impiegati soprattutto per
↳chi lo esercita o l'ha esercitato in passato </description>
  <pubDate>Sat, 15 Sep 2018 08:31:47 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:yRrqpLjBEeiTXpM2dYVoSQ</guid>
  <link>https://hypothes.is/a/yRrqpLjBEeiTXpM2dYVoSQ</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018

```

(continues on next page)

(continua dalla pagina precedente)

```

↪ForumPA documentazione</title>
  <description>&lt;blockquote&gt;formazione puntuale &lt;/blockquote&gt;La
↪formazione puntuale deve inserirsi in un periodo di lavoro dove non si siano vacanze
↪oppure prolungarsi di almeno 6 mesi per quella on line, corsi ad hoc e ben studiati
↪per i lavoratori over 50 anni che hanno difficoltà di apprendimento e di memoria oltre
↪al tempo che si deve dedicare alla famiglia e alle proprie condizioni fisiche </
↪description>
  <pubDate>Sat, 15 Sep 2018 08:28:00 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:QgUg1LjBEeitlevRH5hrFA</guid>
  <link>https://hypothes.is/a/QgUg1LjBEeitlevRH5hrFA</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

<item>
  <title>2. Nuovi processi per la PA abilitante - Libro bianco innovazione 2018
↪ForumPA documentazione</title>
  <description>Sono d&#39;accordo di tenere conto dell&#39;esperienze maturate in
↪ambito lavorativo e dell&#39;anzianità di servizio oltre ad un riconoscimento di CFU a
↪livello di laurea per chi ha superato i 30 anni di servizio grazie </description>
  <pubDate>Sat, 15 Sep 2018 08:16:30 -0000</pubDate>
  <guid isPermaLink="false">tag:hypothes.is,2015-09:pq3Snri_EeiP_FOnvElAKA</guid>
  <link>https://hypothes.is/a/pq3Snri_EeiP_FOnvElAKA</link>
  <dc:creator><![CDATA[LyLilly]]></dc:creator>
</item>

</channel>
</rss>

```

C'è anche in formato JSON:

[url-di-readthedocs / pagina-specifica-del-documento-readthedocs.html](url-di-readthedocs/pagina-specifica-del-documento-readthedocs.html)

Sono diversi modi per seguire i commenti sulle pagine di Read the Docs, e per attivare eventuali notifiche automatiche.

## 4.10.2 Funzionalità di collaborazione per le annotazioni di Hypothes.is

A questo sono illustrate le funzionalità che abilitano alla collaborazione attraverso le annotazioni di Hypothes.is. Si tratta degli “**open group**” e “**restricted group**”.

In sostanza ora sono disponibili:

- Public Layer
- Open Group
- Restricted Group
- Private Group

	Public Layer	Open Group	Restricted Group	Private Group
Who can read annotations?	Anyone	Anyone	Anyone	Only logged-in group members
Who can post annotations?	Any logged-in user	Any logged-in user	Only logged-in group members	Only logged-in group members
Who can join?	N/A as anyone who is logged in to Hypothesis can annotate in the Public Layer	N/A as anyone who is logged in to Hypothesis can annotate in an Open Group	Invite only	Invite only: Group creator can share a link for users to join group

The table above describes the configurations and permissions available to readers and annotators of the Public Layer and Open, Restricted, and Private Groups in Hypothesis.

## 4.11 Sintassi del linguaggio reStructuredText (RST)

Questa pagina è un bonus nel contesto del tutorial. Vuole essere un supporto a coloro che intendono approfondire la conoscenza del linguaggio di programmazione **reStructuredText (RST)**. Di seguito sono elencati alcuni strumenti (anche online) per facilitare la conversione di testo nel linguaggio RST.

Il RST è un **markup language**, un linguaggio di marcatura, che permette di descrivere i dati e le istruzioni attraverso una formattazione specifica che utilizza i cosiddetti tag, cioè dei marcatori.

---

### **Suggerimento: Specifiche del linguaggio reStructuredText (RST). Guide, editor e convertitori online**

- guida alla sintassi del linguaggio RST, utile per capire nel dettaglio come funziona;
  - tool online per convertire il linguaggio Markdown nel linguaggio RST e convertitore per i più diffusi linguaggi di programmazione;;
  - editor di tabelle per più linguaggi di programmazione, per esempio con Markdown;
  - editor di testo per linguaggio RST; → link deprecato
  - , aspetti in comune tra il linguaggio RST e Markdown;
  - (by Sphinx), breve introduzione ai concetti e sintassi del linguaggio reStructuredText (RST), per fornire agli utenti informazioni esaustive per creare produzione di documenti online.
- 

a cura di **Eric Holscher** del

## 4.12 Sintassi del linguaggio Markdown (MD) per le finalità di ‘Read the Docs’

Questa pagina illustra alcune guide online per i requisiti che i file di tipo **Markdown** .MD devono possedere al fine di compilare le ‘builds’ su Read the Docs.

Infatti oltre al formato ReStructuredText .RST anche il formato Markdown .MD può compilare le ‘builds’ su Read the Docs per la visualizzazione dei contenuti nelle pagine HTML.

**Nota:** Questa pagina di tutorial nasce dall’esperienza vissuta nel costruire il tutorial , una guida “*per rispondere alle numerose richieste di aiuto sull’uso del calcolatore di campi e per colmare un vuoto sulla guida online di QGIS con esempi e molti screenshot*”.

Sintassi di base del linguaggio Markdown	
Basic writing and formatting syntax. Create sophisticated formatting for your prose and code on GitHub with simple syntax	
Una breve guida al linguaggio Markdown	
Un tool online per editare codice in Markdown	
Una guida Read the Docs sul linguaggio Markdown	(con utili analisi comparative tra sintassi del linguaggio Markdown e HTML)
Project documentation with Markdown	
markdown-it demo	

### 4.12.1 Fare leggere i file in formato “.md” a Read the Docs

Come descritto in questa , si possono fare leggere file in formato “.md” a Read the Docs.

Azioni da effettuare:

1. bisogna creare un file `requirements.txt` che ha il seguente contenuto:

```

1 sphinx-rtd-theme
2 sphinx
3 recommonmark
4 markdown
5 sphinx-markdown-tables
```

2. bisogna aggiungere nel file `conf.py` le seguenti istruzioni.

Dopo `import sys`, os inserire il seguente codice:

```

1 import recommonmark
2 from recommonmark.transform import AutoStructify
```

(continues on next page)

(continua dalla pagina precedente)

```

3 # Add any paths that contain templates here, relative to this directory.
4 templates_path = ['_templates']
5 html_static_path = ['static']
6 def setup(app):
7     # overrides for wide tables in RTD theme
8     app.add_stylesheet('theme_overrides.css') # path relative to static
9
10 app.add_stylesheet('theme_overrides.css') # path relative to static
11 source_parsers = {
12     '.md': 'recommonmark.parser.CommonMarkParser',
13 }
14
15 source_suffix = ['.rst', '.md']
16
17 extensions = ['sphinx.ext.ifconfig', 'sphinx_markdown_tables']

```

source\_suffix = '.rst' si trasformerà in source\_suffix = ['.rst', '.md']

## 4.13 Pubblicare con il design di Docs Italia

Se una Pubblica Amministrazione vuole sfruttare il servizio del componente aggiuntivo di Google doc, GGeditor, per pubblicare un documento su Read the Docs ma con il design di , ecco alcune cose da tenere in considerazione al fine di non commettere errori che - alla fine - potrebbero non fare compilare la costruzione (build) del documento su Read the Docs e quindi pregiudicare la visualizzazione delle pagine HTML.

The screenshot shows the Docs Italia website interface. At the top, there is a navigation bar with links for 'Piano Triennale', 'Developers', 'Designers', 'Forum', 'Docs', and 'GitHub'. The main header features the 'Docs Italia' logo and the tagline 'Documenti pubblici, digitali.'. The main content area displays the title 'Linee guida open data comune Palermo' and a 'Modifica su GitHub' button. Below the title, there are logos for the City of Palermo, the Italian Government, and the 'Open DATA' initiative. A brief description of the document is provided at the bottom of the page.

Docs Italia parte dallo stile semplice di Read the Docs ma viene arricchito, dal Team Trasformazione Digitale, di “header” (parte superiore della pagina) e “footer” (parte inferiore della pagina) nella visualizzazione delle pagine html e di alcune novità in termini di personalizzazioni nella visualizzazione di alcuni contenuti testuali.

Ci sono differenze nel file `conf.py` del progetto con lo stile basic **Read the Docs** rispetto al file `conf.py` del progetto con il design **Docs Italia** del Team Trasformazione Digitale. Basta conoscerle e si eviteranno errori. Il file `conf.py` sul progetto Github è il cuore della configurazione che permette la corretta visualizzazione delle pagine html e la visualizzazione degli aggiornamenti fatti nelle pagine Google doc!

- un esempio di file `conf.py` del progetto con lo stile di base **Read the Docs**.
- un esempio di file `conf.py` del progetto con lo stile **Docs Italia**.

Evidente la differenza di istruzioni date nei due file.

Di seguito alcuni accorgimenti da tenere presenti nel lavoro da fare.

#### 4.13.1 Uso corretto degli apici e doppi apici nel file ‘conf.py’

Nel file `conf.py` che si trova nel repository di Github, i doppi apici (“”) vanno solo nel primo campo:

```
settings_project_name = "Nome progetto"
```

Nel resto dei settings vanno solo i singoli apici (‘’).

Su `setting_doc_version` e su `setting_doc_release` va la dicitura `'version: latest'` (dentro apici singoli).

```
settings_copyright_copyleft = 'Comune di ...'
settings_editor_name = 'Comune di ...'
settings_doc_version = 'version: latest'
settings_doc_release = 'version: latest'
settings_basename = 'nome_progetto'
settings_file_name = 'nome_progetto'
```

#### 4.13.2 Settaggi del file ‘conf.py’ nella sezione “Options for LaTeX output”

Nella sezione `options for LaTeX output` dare le istruzioni seguendo esattamente questo codice di seguito riportato:

```
# -- Options for LaTeX output -----

latex_elements = {
# The paper size ('letterpaper' or 'a4paper').
'papersize': 'a4paper',

# The font size ('10pt', '11pt' or '12pt').
'pointsize': '10pt',

# Additional stuff for the LaTeX preamble.
'preamble': "",
}
```

Praticamente l'istruzione 'papersize': 'a4paper', **non deve** avere davanti il simbolo del cancelletto (#).

### 4.13.3 Uso corretto dei titoli dei file Google doc dentro il toctree del file index

Nell'editing del nome dei file dei capitoli sul toctree, nel Google doc dell'index, deve essere inserito il suffisso `.rst`. Senza l'aggiunta del suffisso, sulle pagine html dello stile Docs Italia non comparirà la struttura dell'indice in home page.

Nel caso della pubblicazione su Read the Docs **versione basic**, (cioè senza il design Docs Italia elaborato dal Team Digitale per i documenti della PA) l'assenza di questo suffisso `.rst` per ogni file elencato nel toctree del file 'index' non costituisce un problema e l'indice viene visualizzato ugualmente sulle pagine html di Read the Docs.

Esempio di indice nel Google doc "index" nel caso di progetto **Read the Docs**: si nota l'assenza del suffisso `.rst` ad ogni titolo dei file elencati nel toctree

```
.. toctree::
   :maxdepth: 3
   :caption: Indice

   gdocs-rtd
   tutorial
   come-usarlo
   lavoro-github
   lavoro-rtd
   user-guide
   hypothesis-partecipazione
   pubblicare-su-docs-italia
   licenza
```

Esempio di indice nel Google doc "index" nel caso di progetto **Docs Italia**: si nota la presenza del suffisso `.rst` ad ogni titolo dei file elencati nel toctree

```
.. toctree::
   :maxdepth: 3
   :caption: Indice dei contenuti

   CARTA-SERVIZI-BIBLIOTECA-capitolo-1.rst
   CARTA-SERVIZI-BIBLIOTECA-capitolo-2.rst
   CARTA-SERVIZI-BIBLIOTECA-capitolo-3.rst
   CARTA-SERVIZI-BIBLIOTECA-appendice.rst
```

#### 4.13.4 HTML Logo

Sui documenti pubblicati con il design Docs Italia non c'è la possibilità di fare visualizzare un logo/immagine in alto a sinistra come, invece, avviene con la versione Read the Docs basic.

Quindi sul file `conf.py` l'istruzione seguente

```
html_logo = "images/logo.png"
```

non deve essere data in questo caso. Se viene data la compilazione su Read the Docs fallisce.

Può essere editato cancellato prima:

```
# html_logo = "images/logo.png"
```

così facendo l'istruzione non ha effetto in quanto tutto ciò che viene dopo cancellato sul file `conf.py` rappresenta un testo di commento e non un'istruzione da eseguire.

#### 4.13.5 Un 'progetto tipo' da clonare per la pubblicazione con il design Docs Italia

A titolo di , da clonare su Github, per l'esigenza di creazione di un nuovo progetto di pubblicazione con il design **Docs Italia**, può essere usato questo repository su Github: dove sono state effettuate le necessarie verifiche nel file `conf.py` che permette un'esatta compilazione del progetto sul design Docs Italia, ottenendo lo status verde di **passing** .

- Qui il file `conf.py` = .
- Qui i dettagli dell'ultima compilazione del progetto in esame sulla piattaforma Read the Docs = .

E qui di seguito gli unici campi da personalizzare nel file `conf.py`:

```
settings_project_name = "cambiami_nome"
settings_copyright_copyleft = 'Comune di ...'
settings_editor_name = 'Comune di ...'
settings_doc_version = 'version: latest'
settings_doc_release = 'version: latest'
settings_basename = 'cambiami_nome'
settings_file_name = 'cambiami_nome'
```

Se sul sito Read the Docs avete dato, ad esempio, al progetto il titolo **"linee guida open data comune vattelapesca"**, allora nel campo `settings_basename` e nel campo `settings_file_name` date lo stesso nome così:

```
settings_basename = 'linee-guida-open-data-comune-vattelapesca'
settings_file_name = 'linee-guida-open-data-comune-vattelapesca'
```

#### Un ultimo consiglio per semplificare il lavoro

Cercate di dare lo stesso nome al progetto su Github, nel file `conf.py` di Github, e al progetto Read the Docs:

- su Github il nome del progetto da creare = **github/nome\_account/linee guida open data comune vattelapesca**
- sul file `conf.py` di Github, rispettivamente nei due campi `settings_basename` e `settings_file_name` = **linee-guida-open-data-comune-vattelapesca**
- su Read the Docs = **linee guida open data comune vattelapesca**

Questo vi consentirà di avere informazioni omogenee sui diversi ambienti dove andrete a lavorare, eliminando elementi che possono essere causa di errori o di confusione.

---

### 4.13.6 Aggiornamento 2020

AGID e il Dipartimento Trasformazione Digitale del Ministero Innovazione Tecnologica e Digitalizzazione hanno stabilito che:

- è la versione finale della piattaforma di pubblicazione della documentazione della Pubblica Amministrazione;
- la piattaforma può essere usata solamente per la pubblicazione di documenti prodotti dall'AGID e dal Ministero Innovazione Tecnologica e Digitalizzazione.

Ciò significa che qualsiasi altra Pubblica Amministrazione non può ospitare i propri documenti sulla piattaforma . Quindi qualunque PA può usare Read the Docs per pubblicare documentazione e manuali online.

## 4.14 Licenza, chi sono e ringraziamenti

### 4.14.1 Licenza per questo tutorial

Questo tutorial viene rilasciato con licenza . Chiunque può utilizzarne e riusarne i contenuti con la sola condizione di citare l'autore.

### 4.14.2 Chi sono e perché nasce questo tutorial

Sono , dipendente del Comune di Palermo impegnato su e digitalizzazione e innovazione di processi e servizi.

Civic hacker () con la passione della condivisione della conoscenza.

Mi piace moltissimo e quale stile di piattaforma per la pubblicazione di documenti online, ed è il motivo per il quale ho realizzato questo *tutorial*.

### 4.14.3 Iniziate le pubblicazioni su Read the Docs

di pubblicazioni realizzate sullo stile di Read the Docs, molte delle quali in collaborazione con altre persone che condividono la cultura dell'architettura dell'informazione del documento.

### 4.14.4 Un ringraziamento al creatore del componente aggiuntivo GGeditor, Hsin Yuan Yeh



#### GGeditor

Hsin Yuan Yeh

Generate content in reStructuredText format and commit to Github repository.

espone le funzioni svolte dal **plugin** abbastanza bene. L'inglese adottato nel tutorial per le spiegazioni è anche di facile comprensione in quanto scritto da un bravo sviluppatore di Taiwan (**Hsin Yuan Yeh**).

Ha fatto un gran lavoro facilitando la vita a molte persone che non hanno la conoscenza del linguaggio RST, né la pazienza per apprenderlo.

Ha risolto alcuni problemi da me segnalati, che sono descritti nell'homepage di questo tutorial nei box "aggiornamenti".

### 4.14.5 Un messaggio di errore "undefined", facile da risolvere

**Hsin Yuan Yeh** ha fornito l'assistenza durante alcune sessioni di utilizzo del suo plugin nelle quali ho incontrato qualche problema di funzionamento (un messaggio di "undefined" e un messaggio di error in lingua taiwanese) nella procedura di commit del Google Doc a Github.

Ho aperto una issue () sul suo progetto Github e Hsin Yuan Yeh ha provveduto ad analizzare tempestivamente il problema. Ha fornito immediatamente una soluzione che è stata rilasciata da Google con una seconda release del componente aggiuntivo **GGeditor**.

Praticamente se dovesse capitare di leggere un messaggio "undefined" o un messaggio in lingua taiwanese del tipo

↓

無法讀取 null 的  
[ 19rZn4reioTxYQL2dLiVbxOHRMnw2llq\_5r6xtLpPqVQ ] 屬  
性。

OK

la cosa da fare è: andare in “Componenti aggiuntivi”, poi “GGeditor” e quindi su “Setting” dove si trova un tasto rosso di “Reset” che cancella tutte le informazioni di collegamento agli account Github e ai relativi file nel repository.

## Settings

Accounts

Document

About

The "Reset" button will clean up all your stored information of accounts and bindings.

RESET

L'ho provato e tutto ritorna in ordine e funziona correttamente nella procedura di Commit da Google Doc a Github. Ovviamente bisogna ricollegare il componente aggiuntivo GGeditor all'account di Github per potere continuare a effettuare i commits.

Grazie Hsin Yuan Yeh per questa tempestiva soluzione al problema!

### 4.14.6 GGeditor, un servizio online da utilizzare per la pubblicazione di documenti delle Pubbliche Amministrazioni

Mi auguro che la semplicità d'uso di questo strumento descritto nel tutorial possa stimolare i dipendenti e dirigenti pubblici, partendo da un Google Doc, a pubblicare documenti su Read the Docs e con lo stile Docs Italia, al posto degli scomodi PDF. Per chi fosse interessato ad utilizzare questo strumento, sono disponibile a fornire aiuto. Il mio contatto è:

## 4.15 Opendatasicilia

---

**Suggerimento:** il contenuto di questa pagina che stai leggendo è editato in ... dai un'occhiata per capire meglio come il testo di Google doc viene esposto su pagine HTML di *Read the Docs*

---