
Conversione del Codice dell'amministrazione digitale in formato Read the Docs

Release version: latest

2018, AgID, Team Digitale

06 feb 2018

1	Panoramica del processo di conversione	3
1.1	Creazione dei testi di input da Normattiva	3
1.2	Conversione in formato reStructuredText	3
1.3	Revisione manuale	5
1.4	Divisione del CAD nei singoli file	5
1.5	Caricamento delle versioni nel repo locale	6
1.6	Caricamento dei file sul repo Italia	6

Il Codice dell'amministrazione digitale (CAD), emanato con il Decreto Legislativo 7 marzo 2005, n. 82, ha subito diverse revisioni nel corso degli anni. Il testo completo delle varie versioni è disponibile all'indirizzo cad.readthedocs.io.

Sul [sito di Normattiva](#) è possibile visualizzare i testi del CAD nel corso della sua storia, tramite gli strumenti di ricerca avanzata. Il [Team per la Trasformazione Digitale](#) ha convertito le diverse versioni in formato testuale e le ha organizzate con un sistema a rilasci successivi, tipico dello sviluppo del software, sfruttando le potenzialità di [GitHub](#), [Sphinx](#) e [Read the Docs](#).

Il risultato è un sito che contiene tutte le revisioni del CAD, accessibili tramite una barra laterale e facilmente navigabili. Chi desidera leggere il Codice può ora visualizzare l'evoluzione di ciascun Articolo, analizzando, se desidera, le modifiche apportate riga per riga.

Questo tutorial spiega come è stato realizzato il sito e quali problemi sono stati affrontati durante il processo.

Panoramica del processo di conversione

1.1 Creazione dei testi di input da Normattiva

Come primo passo, è necessario ottenere i testi del CAD da [Normattiva](#). Gli strumenti di ricerca del sito permettono di ottenere le versioni di una determinata legge in vigore in una data specifica. La [Figura 1.1](#) mostra il motore di ricerca avanzato di Normattiva. Selezionando la **Data di vigenza** opportuna, è possibile ottenere le diverse versioni del CAD nel corso della sua storia. Normattiva permette di selezionare date nel futuro, qualora una nuova versione di una legge sia stata approvata ma non ancora entrata in vigore.

Il procedimento è laborioso, perché non esiste una funzione API che permetta un accesso programmatico a tutto il testo di un documento legislativo. Una volta trovata la versione richiesta di una legge, è possibile visualizzarne il testo completo tramite il pulsante **Versione stampabile** in alto a sinistra. La [Figura 1.2](#) mostra il testo del CAD in vigore il 24 gennaio 2018.

Tutte le versioni del CAD ottenute da Normattiva.it sono contenute nella cartella *input* in questo repo.

1.2 Conversione in formato reStructuredText

Ottenuti i testi delle versioni del CAD, è possibile eseguire una prima conversione in [formato reStructuredText \(RST\)](#). Questo è il formato utilizzato dal tool [Sphinx](#) per produrre la documentazione.

Lo script `normattiva2rst.py` è un programma in Python che accetta un file di testo in input e restituisce un file convertito in formato RST. La sintassi per l'uso dello script è:

```
python3 normattiva2rst.py source-file > output.rst
```

Le funzioni principali dello script sono:

1. Correggere la grafia delle parole accentate, che nei testi di Normattiva appaiono sempre con l'apostrofo.
2. Rimuovere le doppie parentesi ((...)), che in Normattiva denotano una eliminazione di testo dalla legge.
3. Riconoscere *Capi*, *Sezioni* e *Articoli* e creare le decorazioni per i titoli, secondo una precisa gerarchia. A Capi e Sezioni sono assegnati rispettivamente titoli di primo e secondo livello, mentre gli Articoli vengono

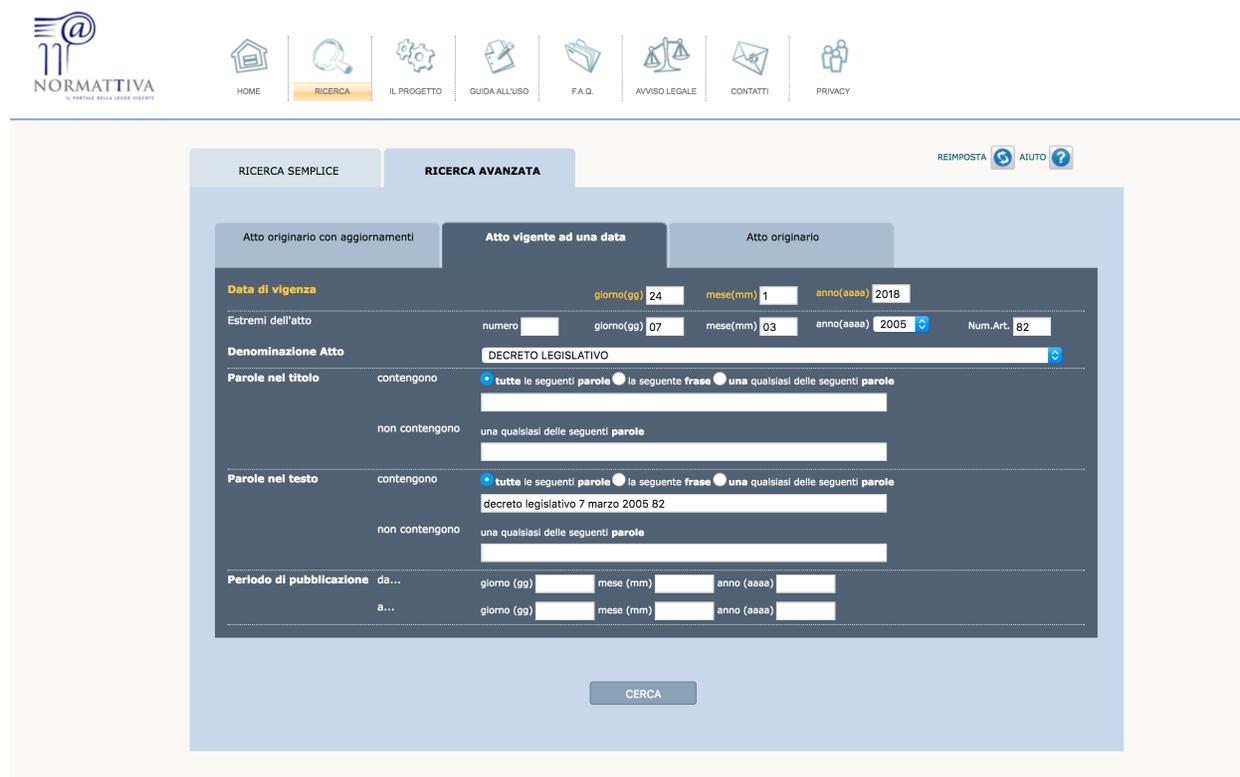


Fig. 1.1: Il motore di ricerca avanzato di Normattiva.

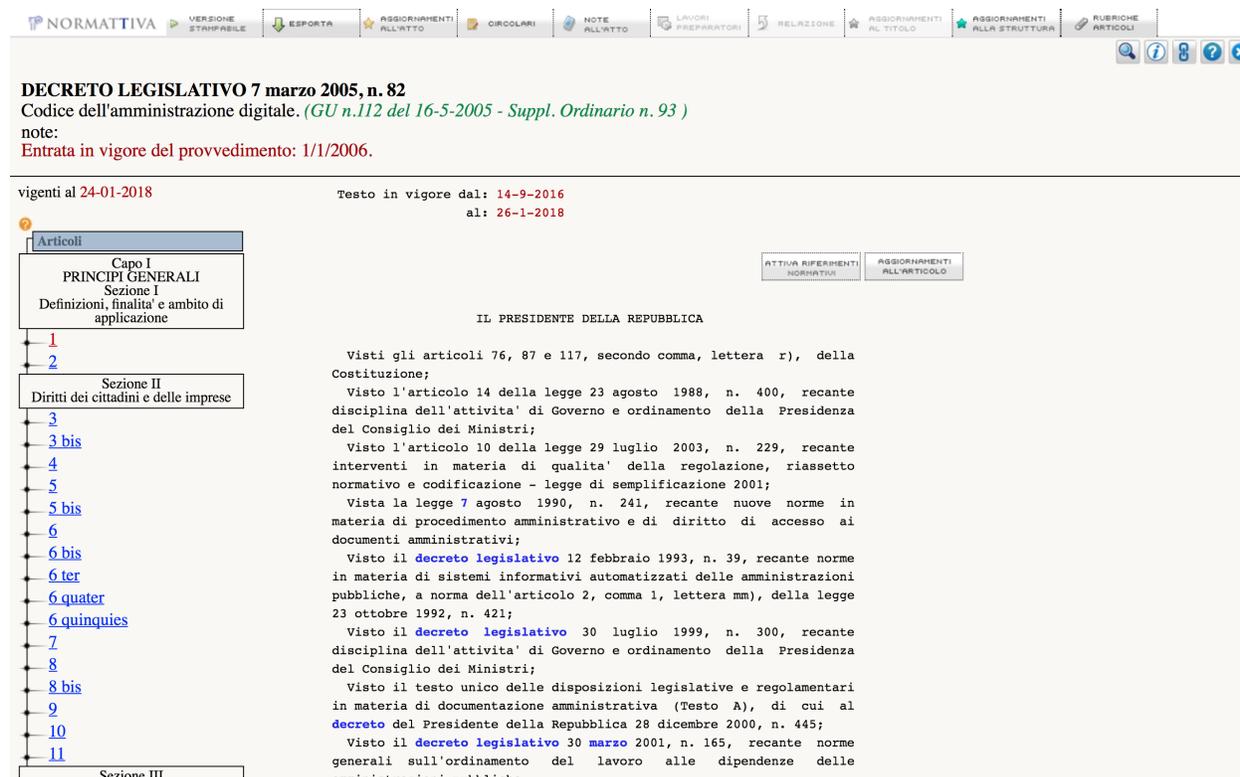


Fig. 1.2: Il testo del CAD come visualizzato su Normattiva.

contrassegnati in grassetto. Agli articoli non viene assegnata una decorazione per il titolo per evitare problemi nella gerarchia: la sintassi RST non permette a un titolo di terzo livello (un Articolo) di seguire un titolo di primo livello (un Capo), come invece avviene all'interno del CAD.

4. Indentare i commi e i paragrafi in maniera consistente con la sintassi RST.

Il processo di conversione delle varie versioni del CAD è stato automatizzato con uno script in Bash, `bulk-text2rst.sh`, che applica lo script `normattiva2rst.py` a tutti i testi contenuti nella cartella `input`.

1.3 Revisione manuale

I testi contenuti nella cartella `rst` devono essere corretti manualmente prima di una ulteriore conversione.

Nel testo di `Normattiva`, infatti, a volte sono presenti degli errori o delle inconsistenze che impediscono a `normattiva2rst.py` di effettuare la conversione correttamente.

Per esempio:

- un comma in cui il numero iniziale non è separato dal testo seguente tramite uno spazio («1.Il testo» invece che «1. Il testo») non viene riconosciuto come comma separato;
- a volte, `Normattiva` presenta il titolo degli articoli fra parentesi e con carattere minuscolo. Anche in questo caso è necessario intervenire per correggere l'errore;
- in alcuni casi, le doppie parentesi ((...)) sono accompagnate da spazi extra. Quando queste vengono rimosse, quindi, rimane uno spazio in più che deve essere rimosso.

È possibile che queste correzioni possano essere automatizzate all'interno dello script in futuro, grazie all'inserimento di test più raffinati nel codice.

Per ottimizzare i tempi ed evitare la propagazione di errori fra le varie versioni, è possibile sfruttare la funzione `diff` di molti editor di testo per correggere solo le parti differenti fra una versione e la successiva. In questo modo, è necessaria una revisione approfondita soltanto della prima versione del CAD. Gli errori eventualmente presenti nelle successive versioni verranno evidenziati dal `diff` e potranno essere corretti.

1.4 Divisione del CAD nei singoli file

In questa fase, i testi contenuti nella cartella `rst` vengono sottoposti a un ulteriore trattamento tramite lo script `splitRST-softwrap.py`. Per ciascuna versione, lo script esegue le seguenti operazioni:

1. Creare una cartella con le relative sottocartelle in cui salvare i file.
2. Leggere la struttura del testo, individuando Capi, Sezioni e Articoli e salvandoli in array differenti.
3. Rimuovere gli «a capo» all'interno di ciascun Articolo, unendo righe successive tramite spazio.
4. Per ciascun Articolo individuato, creare un file con un nome che dipende dal Capo ed eventualmente dalla Sezione in cui si trova.
5. Creare il file `index.rst`, includendo il titolo del documento e i link a tutti i Capi.
6. Per ciascun Capo, creare un file con tutti i link alle Sezioni o direttamente agli Articoli contenuti.
7. Per ciascuna Sezione, creare un file con tutti i link agli Articoli contenuti.

Quando lo script termina l'esecuzione, ciascuna cartella all'interno di `rst` contiene i file nella versione corretta per creare la documentazione in [Read the Docs](#).

1.5 Caricamento delle versioni nel repo locale

Prima di caricare la successione delle versioni del CAD, è necessario creare un repo contenente i file di base per la compilazione dei documenti con Sphinx. La struttura è quella della cartella *repo-base* contenuta in questo repo. Una volta salvati i file ed effettuato un commit, è possibile cominciare.

Il caricamento successivo delle varie versioni del CAD può essere nuovamente automatizzato con uno script Bash.

La struttura dello script è la seguente:

```
#!/bin/bash

for NAME in $( ls output); do
cd percorso-assoluto-a/repo-base
cp -R percorso-assoluto-a/output/"$NAME"/* .
git add *
git commit -m $NAME
git tag $NAME
done
```

Per ciascuna cartella contenuta nella cartella *output*, lo script si assicura di operare all'interno del repo, poi copia tutti i file relativi a una versione nel repo. Infine, vengono effettuate le classiche operazioni di Git: *add*, *commit* e *tag*.

1.6 Caricamento dei file sul repo Italia

Come ultima operazione, è necessario caricare tutto il branch con la storia del CAD nel repo remoto di github.com/Italia. L'unico punto a cui fare attenzione è non dimenticare i tag! Questi, infatti, non vengono sincronizzati di default con il repo remoto, ma l'utente deve aggiungere il comando `--tags`.

Il comando completo per eseguire il *push* è simile a: `git push origin master --tags`

Infine, è possibile creare un progetto su [Read the Docs](https://readthedocs.org/). L'intero progetto del Codice dell'amministrazione digitale, convertito tramite il procedimento illustrato in questo tutorial, è disponibile sul sito: <http://cad.readthedocs.io>.