

---

# Heritage.py Documentation

*Release 0.1.1*

**Hrishikesh Terdalkar**

**Jun 11, 2023**



**CONTENTS:**

<b>1</b>	<b>Features</b>	<b>3</b>
1.1	Heritage.py . . . . .	3
1.2	Installation . . . . .	4
1.3	Usage . . . . .	5
1.4	heritage . . . . .	5
1.5	Contributing . . . . .	12
1.6	Credits . . . . .	15
1.7	History . . . . .	15
<b>2</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



Heritage.py is a python interface to [The Sanskrit Heritage Site](#).

- Free software: GNU General Public License v3
- Documentation: <https://heritage-py.readthedocs.io>.



## FEATURES

- Morphological Analysis
- Sandhi Formation
- Declensions
- Conjugations

### 1.1 Heritage.py

Heritage.py is a python interface to [The Sanskrit Heritage Site](#).

- Free software: GNU General Public License v3
- Documentation: <https://heritage-py.readthedocs.io>.

#### 1.1.1 Features

- Morphological Analysis
- Sandhi Formation
- Declensions
- Conjugations

### 1.1.2 Install

To install Heritage.py, run this command in your terminal:

```
$ pip install heritage
```

### 1.1.3 Usage

Heritage.py has two possible modes of operation,

1. Using a web mirror

This mode uses any compatible web mirror of The Heritage Platform (e.g. <https://sanskrit.inria.fr/index.en.html>) and does not require any installation, however, HTTP requests are made for every task resulting in a larger delay.

2. Using a local installation

**Installation Instructions:** <https://sanskrit.inria.fr/manual.html#installation>.

This mode requires a local installation of The Heritage Platform. As a result, it is considerably faster in obtaining results.

To use Heritage.py in a project,

```
import heritage
```

### 1.1.4 Credits

This package was created with [Cookiecutter](#) and the [hrishikeshrt/cookiecutter-pypackage](#) project template.

## 1.2 Installation

### 1.2.1 Stable release

To install Heritage.py, run this command in your terminal:

```
$ pip install heritage
```

This is the preferred method to install Heritage.py, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

### 1.2.2 From sources

The sources for Heritage.py can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/hrishikeshrt/heritage
```

Or download the [tarball](#):



```
$ curl -OJL https://github.com/hrishikeshrt/heritage/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

## 1.3 Usage

Heritage.py has two possible modes of operation,

1. Using a web mirror

This mode uses any compatible web mirror of The Heritage Platform (e.g. <https://sanskrit.inria.fr/index.en.html>) and does not require any installation, however, HTTP requests are made for every task resulting in a larger delay.

2. Using a local installation

**Installation Instructions:** <https://sanskrit.inria.fr/manual.html#installation>.

This mode requires a local installation of The Heritage Platform. As a result, it is considerably faster in obtaining results.

To use Heritage.py in a project,

```
import heritage
```

## 1.4 heritage

### 1.4.1 heritage package

#### Submodules

#### heritage.cli module

Console script for Heritage.

`heritage.cli.main()`

Console script for Heritage.py

#### heritage.constants module

Constants

## heritage.heritage module

Python Interface to The Sanskrit Heritage Site

Use The Sanskrit Heritage Platform using,

- Web mirror - no installation required - makes HTTP requests
- Local installation - faster - uses console - no HTTP requests required

## Using Local Installation

- Heritage\_Platform/ML/ contains the scripts
- export QUERY\_STRING as shell variable (referred to as OPTION\_STRING in this code alongwith the '&text=TEXT' part)
- execute various scripts, such as ./reader
- still produces HTML output that needs to be parsed

# Default input needs to be in the devanagari format # utils.devanagari\_to\_velthuis() function will convert this to VH

**class** heritage.heritage.frozendict

Bases: dict

heritage.heritage.freezeargs(*func*)

Transform mutable dictionary arguments into immutable frozen ones

Useful to be compatible with @cache. Should be added on top of @cache

heritage.heritage.timeout\_handler(*signum, frame*)

**class** heritage.heritage.HeritageAnalysis(*case: str = None, number: str = None, gender: str = None, tense: str = None*)

Bases: object

**case:** str = None

**number:** str = None

**gender:** str = None

**tense:** str = None

**class** heritage.heritage.Token

Bases: object

**class** heritage.heritage.HeritageOutput(*html: str*)

Bases: object

Heritage Output Parser

Parse output generated by various utilities from Heritage Platform

**CLASSES** = {'footer': ['enpied']}

**process**(*html: Optional[str] = None*)

Process the html and extract basic information

**extract\_analysis**(*meta: bool = False*)

Extract analysis from HTML

**Parameters**

**meta** (*bool*) – If True, include meta information, i.e, parse options, classes The default is False.

**extract\_parse**()

Extract parse from HTML

**extract\_declensions**(*headers: bool = True*)

Extract declensions from HTML

**extract\_conjugations**(*headers: bool = True*)

Extract conjugations from HTML

**extract\_sandhi**()

Extract Sandhi from HTML

**extract\_lexicon\_entry**(*word\_id: str*)

Extract entry from a lexicon

**static parse\_analysis**(*table: Tag*)

Parse analysis of a single word Analysis Format is: [root]{analysis\_1 | analysis\_2 | ..}

**Parameters**

**table** (*bs4.element.Tag*) – Valid *table* element

**Returns**

**analysies**

**Return type**

list

**class** heritage.heritage.**HeritagePlatform**(*base\_dir: str = "", base\_url: Optional[str] = None, method: str = 'shell', \*\*kwargs*)

Bases: object

The Sanskrit Heritage Platform

Access various utilities from The Sanskrit Heritage Platform

Initialize Heritage Class

**Parameters**

- **base\_dir** (*str*) – Path to the Heritage\_Platform repository. The directory should contain 'ML' sub-directory, which further contains the scripts
- **base\_url** (*str, optional*) – URL for the Heritage Platform Mirror. If None, the official INRIA website will be used. The default is None.
- **method** (*str, optional*) – Method used to obtain results. Results can be obtained either using the web installation or using UNIX shell.

Possible values are, 'shell' and 'web' The default is 'shell'.

**INRIA\_URL** = 'https://sanskrit.inria.fr/cgi-bin/SKT/'

```
ACTIONS = {'conjugation': {'shell': 'conjugation', 'web': 'sktconjug.cgi'},
'declension': {'shell': 'declension', 'web': 'sktdeclin.cgi'}, 'dictionary':
{'shell': '../MW/', 'web': '../..../MW/'}, 'interface': {'shell': 'interface',
'web': 'sktgraph.cgi'}, 'lemma': {'shell': 'lemmatizer', 'web':
'sktlemmatizer.cgi'}, 'parser': {'shell': 'parser', 'web': 'sktparser.cgi'},
'reader': {'shell': 'reader', 'web': 'sktreader.cgi'}, 'sandhi': {'shell':
'sandhier', 'web': 'sktsandhier.cgi'}, 'search': {'shell': 'indexer', 'web':
'sktindex.cgi'}, 'search_easy': {'shell': 'indexerd', 'web': 'sktsearch.cgi'},
'user': {'shell': 'user_aid', 'web': 'sktuser.cgi'}}
```

```
OPTIONS = {'font': {'default': 'deva', 'description': 'Font for Sanskrit output',
'values': {'deva': 'Devanagari', 'roma': 'Roman (IAST)'}}, 'lex': {'default':
'MW', 'description': 'Lexicon', 'values': {'MW': 'Monier-Williams Dictionary
(English)', 'SH': 'Sanskrit Heritage Dictionary (French)'}}, 't': {'default':
'VH', 'description': 'Internal Transliteration Scheme', 'values': {'VH':
'Velthuis'}}
```

```
METHODS = ['shell', 'web']
```

```
DEFAULT_METHOD = 'shell'
```

```
__init__(base_dir: str = "", base_url: Optional[str] = None, method: str = 'shell', **kwargs)
```

Initialize Heritage Class

#### Parameters

- **base\_dir** (*str*) – Path to the Heritage\_Platform repository. The directory should contain ‘ML’ sub-directory, which further contains the scripts
- **base\_url** (*str*, *optional*) – URL for the Heritage Platform Mirror. If None, the official INRIA website will be used. The default is None.
- **method** (*str*, *optional*) – Method used to obtain results. Results can be obtained either using the web installation or using UNIX shell.

Possible values are, ‘shell’ and ‘web’ The default is ‘shell’.

```
get_analysis(input_text: str, sentence: bool = True, unsandhied: bool = False, meta: bool = False)
```

Obtain morphological analyses using The Sanskrit Reader Companion

#### Parameters

- **input\_text** (*str*) – Input text to analyse
- **sentence** (*bool*, *optional*) – The input is treated as a sentence, if true, otherwise as a word. The default is True.
- **unsandhied** (*bool*, *optional*) – If True, the input text is assumed to not contain sandhi. The default is False.
- **meta** (*bool*, *optional*) – The option is passed to HeritageOutput.extract\_analysis(). The default is False.

#### Returns

Dictionary of valid morphological analyses with solution\_id as keys

#### Return type

dict

**get\_parse**(*input\_text: str, solution\_id: Optional[int] = None, sentence: bool = True, unsandhied: bool = False*)

Obtain parse of a sentence using The Sanskrit Reader Companion

**Parameters**

- **input\_text** (*str*) – Input text to analyse
- **solution\_id** (*int, optional*) – Solution ID to parse. If None, the first solution ID is used. The default is None.
- **sentence** (*bool, optional*) – The input is treated as a sentence, if true, otherwise as a word. The option is passed to HeritagePlatform.get\_analysis(). The default is True.
- **unsandhied** (*bool, optional*) – If True, the input text is assumed to not contain sandhi. The option is passed to HeritagePlatform.get\_analysis(). The default is False.

**Returns**

Parse of the sentence

**Return type**

dict

**sandhi**(*word\_1: str, word\_2: str, mode: str = 'internal'*)

Join two words by forming a Sandhi

**Parameters**

- **word\_1** (*str*) – The first (left) word in the Sandhi
- **word\_2** (*str*) – The second (right) word in the Sandhi
- **mode** (*str, optional*) – Indicates whether the words join to form a single word or not. Possible values are, \* internal \* external. The default is 'internal'.

**Returns**

**sandhi** – String obtained by forming the Sandhi

**Return type**

str

**search\_inflected\_form**(*word: str, category: str*)

Search an inflected form

**Parameters**

- **word** (*str*) – Sanskrit Word to search (in Devanagari)
- **category** (*str*) –

**Type of the word**

- Noun: Noun
- Pron: Pronoun
- Part: Participle
- Inde: Indeclinable
- Absya, Abstvaa, Voca, Iic, Ifc, Iiv, Piic etc.

**Returns**

**matches** – List of matches.

**Return type**

list

**get\_declensions**(*word: str, gender: str, headers: bool = True, lexicon: Optional[str] = None*)**get\_conjugations**(*word: str, gana: str, lexicon: Optional[str] = None*)**search\_lexicon**(*word: str, lexicon: Optional[str] = None*)

Search a word in the dictionary

**Parameters**

- **word** (*str*) – Sanskrit Word to search (in Devanagari)
- **lexicon** (*str, optional*) – Lexicon to search the word in. Possible values are,
  - MW: Monier-Williams Dictionary
  - SH: Heritage DictionaryThe default is ‘MW’.

**Returns****matches** – List of matches.**Return type**

list

**get\_lexicon\_entry**(*file\_name: str, word\_id: str*)**get\_result\_from\_web**(*url: str, options: dict, attempts: int = 3*)

Get results from the Heritage Platform web mirror Exponential backoff is used in case there are network errors

**Parameters**

- **url** (*str*) – URL of the CGI script to call `HeritagePlatform.get_url()` can be used to generate supported URLs
- **options** (*dict*) – Dictionary containing valid options for the script
- **attempts** (*int, optional*) – Number of attempts for the exponential backoff The default is 3.

**Returns**

Result (HTML) obtained

**Return type**

str

**get\_result\_from\_shell**(*path: str, options: dict, timeout: int = 30*)

Get results from the Heritage Platform’s local installation via shell

**Parameters**

- **path** (*str*) – Path to the executable script `HeritagePlatform.get_path()` can be used to generate supported paths
- **options** (*dict*) – Valid options for the script
- **timeout** (*int, optional*) – Timeout in seconds, after which the function will abort. The default is 30.

**Returns****result** – Result (HTML) obtained

**Return type**

str

**get\_result**(*action: str, options: dict, \*args, \*\*kwargs*)

High-level function to obtain result for various actions

Avoids the hassle of generating the URL or PATH. Utilizes the `HeritagePlatform.method` attribute to determine whether to fetch through shell or web.

**Parameters**

- **action** (*str*) – Action value corresponding to the utility to be used. Refer to `HeritagePlatform.ACTIONS`
- **options** (*dict*) – Valid options for the specified action

**Returns**

Result (HTML) obtained

**Return type**

str

**get\_method**()

Get the current method

**set\_method**(*method: str*)

Set method for fetching the output

Valid methods are listed in `HeritagePlatform.METHODS`**get\_option**(*opt\_name: str*)

Get the value of global options

**set\_option**(*opt\_name: str, opt\_value: str*)

Set global options

Any of these options, if expected by a particular utility from the Heritage Platform, will be directly used in the `QUERY_STRING` while fetching the output from that utility

class variable `OPTIONS` stores the default values for options

Each option contains, - a ‘description’ of the option - ‘values’ it can take (and descriptions of those values)  
- ‘default’ value

**get\_font**()

Get current font for Sanskrit Output

**set\_font**(*font: str*)

Set font for Sanskrit output

**get\_lexicon**()

Get current lexicon

**set\_lexicon**(*lexicon: str*)

Set lexicon

**get\_url**(*action: str*)

URL Builder

**get\_path**(*action: str*)

Path Builder

**valid\_installation()**

Check if the Heritage Platform installation exists

**static prepare\_input(input\_text: str)**

**Prepare Input**

- Convert Devanagari to Velthuis
- Join words by '+' instead of by whitespaces

**static identify\_gender(gender: str)**

## heritage.utils module

Utility Functions

**heritage.utils.build\_query\_string(options: dict) → str**

Build QUERY\_STRING

**heritage.utils.devanagari\_to\_velthuis(text: str) → str**

Convert Devanagari text to Velthuis

Heritage Platform uses its own DN to VH conversion This deviates from the standard one (from Wiki or other sources) Following is a translation of the JS function convert() from the Heritage Platform Source URL: <https://sanskrit.inria.fr/DICO/utf82VH.js>

## Module contents

Heritage.py – Python Interface to The Sanskrit Heritage Platform

## 1.5 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

### 1.5.1 Types of Contributions

#### Report Bugs

Report bugs at <https://github.com/hrishikeshrt/heritage/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.



## Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

## Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

## Write Documentation

Heritage.py could always use more documentation, whether as part of the official Heritage.py docs, in docstrings, or even on the web in blog posts, articles, and such.

## Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/hrishikeshrt/heritage/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 1.5.2 Get Started!

Ready to contribute? Here’s how to set up *heritage* for local development.

1. Fork the *heritage* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/heritage.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv heritage
$ cd heritage/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you’re done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 heritage tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

### 1.5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check [https://travis-ci.com/hrishikeshrt/heritage/pull\\_requests](https://travis-ci.com/hrishikeshrt/heritage/pull_requests) and make sure that the tests pass for all supported Python versions.

### 1.5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_heritage
```

### 1.5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

## 1.6 Credits

### 1.6.1 Development Lead

- Hrishikesh Terdalkar <[hrishikeshrt@linuxmail.org](mailto:hrishikeshrt@linuxmail.org)>

### 1.6.2 Contributors

None yet. Why not be the first?

## 1.7 History

### 1.7.1 0.1.0 (2022-03-23)

- First release on PyPI.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### h

- `heritage`, [12](#)
- `heritage.cli`, [5](#)
- `heritage.constants`, [5](#)
- `heritage.heritage`, [6](#)
- `heritage.utils`, [12](#)





## Symbols

`__init__()` (*heritage.heritage.HeritagePlatform* method), 8

## A

`ACTIONS` (*heritage.heritage.HeritagePlatform* attribute), 7

## B

`build_query_string()` (in module *heritage.utils*), 12

## C

`case` (*heritage.heritage.HeritageAnalysis* attribute), 6

`CLASSES` (*heritage.heritage.HeritageOutput* attribute), 6

## D

`DEFAULT_METHOD` (*heritage.heritage.HeritagePlatform* attribute), 8

`devanagari_to_velthuis()` (in module *heritage.utils*), 12

## E

`extract_analysis()` (*heritage.heritage.HeritageOutput* method), 6

`extract_conjugations()` (*heritage.heritage.HeritageOutput* method), 7

`extract_declensions()` (*heritage.heritage.HeritageOutput* method), 7

`extract_lexicon_entry()` (*heritage.heritage.HeritageOutput* method), 7

`extract_parse()` (*heritage.heritage.HeritageOutput* method), 7

`extract_sandhi()` (*heritage.heritage.HeritageOutput* method), 7

## F

`freezeargs()` (in module *heritage.heritage*), 6

`frozendict` (class in *heritage.heritage*), 6

## G

`gender` (*heritage.heritage.HeritageAnalysis* attribute), 6

`get_analysis()` (*heritage.heritage.HeritagePlatform* method), 8

`get_conjugations()` (*heritage.heritage.HeritagePlatform* method), 10

`get_declensions()` (*heritage.heritage.HeritagePlatform* method), 10

`get_font()` (*heritage.heritage.HeritagePlatform* method), 11

`get_lexicon()` (*heritage.heritage.HeritagePlatform* method), 11

`get_lexicon_entry()` (*heritage.heritage.HeritagePlatform* method), 10

`get_method()` (*heritage.heritage.HeritagePlatform* method), 11

`get_option()` (*heritage.heritage.HeritagePlatform* method), 11

`get_parse()` (*heritage.heritage.HeritagePlatform* method), 8

`get_path()` (*heritage.heritage.HeritagePlatform* method), 11

`get_result()` (*heritage.heritage.HeritagePlatform* method), 11

`get_result_from_shell()` (*heritage.heritage.HeritagePlatform* method), 10

`get_result_from_web()` (*heritage.heritage.HeritagePlatform* method), 10

`get_url()` (*heritage.heritage.HeritagePlatform* method), 11

## H

`heritage` module, 12

`heritage.cli` module, 5

`heritage.constants` module, 5

`heritage.heritage`

module, 6  
heritage.utils  
module, 12  
HeritageAnalysis (class in *heritage.heritage*), 6  
HeritageOutput (class in *heritage.heritage*), 6  
HeritagePlatform (class in *heritage.heritage*), 7

## I

identify\_gender() (her-  
itage.heritage.HeritagePlatform static method),  
12

INRIA\_URL (heritage.heritage.HeritagePlatform at-  
tribute), 7

## M

main() (in module *heritage.cli*), 5

METHODS (heritage.heritage.HeritagePlatform attribute),  
8

module  
heritage, 12  
heritage.cli, 5  
heritage.constants, 5  
heritage.heritage, 6  
heritage.utils, 12

## N

number (heritage.heritage.HeritageAnalysis attribute), 6

## O

OPTIONS (heritage.heritage.HeritagePlatform attribute),  
8

## P

parse\_analysis() (heritage.heritage.HeritageOutput  
static method), 7

prepare\_input() (heritage.heritage.HeritagePlatform  
static method), 12

process() (heritage.heritage.HeritageOutput method),  
6

## S

sandhi() (heritage.heritage.HeritagePlatform method),  
9

search\_inflected\_form() (her-  
itage.heritage.HeritagePlatform method),  
9

search\_lexicon() (her-  
itage.heritage.HeritagePlatform method),  
10

set\_font() (heritage.heritage.HeritagePlatform  
method), 11

set\_lexicon() (heritage.heritage.HeritagePlatform  
method), 11

set\_method() (heritage.heritage.HeritagePlatform  
method), 11

set\_option() (heritage.heritage.HeritagePlatform  
method), 11

## T

tense (heritage.heritage.HeritageAnalysis attribute), 6

timeout\_handler() (in module *heritage.heritage*), 6

Token (class in *heritage.heritage*), 6

## V

valid\_installation() (her-  
itage.heritage.HeritagePlatform method),  
11