

---

**notes**

**fangjun**

**May 09, 2022**



## CONTENTS:

<b>1</b>	<b>Sphinx</b>	<b>3</b>
1.1	Setup . . . . .	3
<b>2</b>	<b>git</b>	<b>5</b>
2.1	Commands . . . . .	5
2.1.1	rev-parse . . . . .	5
<b>3</b>	<b>docker</b>	<b>7</b>
3.1	Installation . . . . .	7
3.1.1	macos . . . . .	7
<b>4</b>	<b>LaTeX</b>	<b>9</b>
4.1	TikZ . . . . .	9
4.1.1	Basics . . . . .	9
<b>5</b>	<b>Kaldi</b>	<b>11</b>
5.1	Decoding . . . . .	11
<b>6</b>	<b>Python</b>	<b>13</b>
6.1	asyncio . . . . .	13
6.1.1	Hello World . . . . .	13
6.1.2	References . . . . .	13
6.2	argv . . . . .	13
<b>7</b>	<b>bash</b>	<b>15</b>
7.1	sort . . . . .	15
7.2	echo . . . . .	15
<b>8</b>	<b>CUDA</b>	<b>17</b>
8.1	Installation . . . . .	17
8.1.1	CUDA 10.1.243 . . . . .	17
8.1.2	CUDA 11.0.3 . . . . .	17
8.1.3	CUDA 11.3.1 . . . . .	18
8.1.4	CUDA 11.5.2 . . . . .	18
8.1.5	CUDA 11.6.1 . . . . .	18
<b>9</b>	<b>torch</b>	<b>19</b>
9.1	DDP . . . . .	19
9.1.1	Initialization . . . . .	19
<b>10</b>	<b>java</b>	<b>21</b>

10.1	Install . . . . .	21
10.1.1	formatter . . . . .	21
10.1.2	JDK . . . . .	21
10.2	Hello world . . . . .	22
10.3	Reference . . . . .	23
<b>11</b>	<b>javascript</b>	<b>25</b>
11.1	Hello world . . . . .	25
11.2	TODOs . . . . .	25
<b>12</b>	<b>HTML</b>	<b>27</b>
12.1	Hello world . . . . .	27
12.1.1	comments . . . . .	27
12.1.2	images . . . . .	27
12.1.3	ordered lists . . . . .	27
12.1.4	unordered lists . . . . .	28
12.1.5	links . . . . .	28
12.2	References . . . . .	28
<b>13</b>	<b>css</b>	<b>29</b>
13.1	Hello world . . . . .	29
13.1.1	comment . . . . .	29
13.1.2	Selector . . . . .	29
13.2	References . . . . .	29

Download this website in a single [pdf file](#).



This page describes how this website is setup.

## 1.1 Setup

1. Install the dependencies in `./docs/requirements.txt`.

```
sphinx==4.3.2
sphinx-autodoc-typehints==1.12.0
sphinx_rtd_theme==1.0.0
sphinxcontrib-bibtex==2.4.1
```

2. Use `sphinx-quickstart` to generate the skeleton. When it prompts:

```
Separate source and build directories(y/n)
```

Answer yes.

3. Edit `docs/source/conf.py` and add the following lines to it:

```
import sphinx_rtd_theme
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.autosummary',
    'sphinx.ext.githubpages',
    'sphinx.ext.mathjax',
    'sphinx.ext.napoleon',
    'sphinx.ext.todo',
    'sphinx.ext.viewcode',
    'sphinxcontrib.bibtex',
]

html_theme = 'sphinx_rtd_theme'

master_doc = 'index'
pygments_style = 'sphinx'
html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]
smartquotes = False
html_show_sourcelink = True

html_context = {
```

(continues on next page)

(continued from previous page)

```
'display_github': True,
'github_user': 'csu-fangjun',
'github_repo': 'notes',
'github_version': 'master',
'conf_py_path': '/docs/source/',
}

html_theme_options = {
    'logo_only': False,
    'display_version': True,
    'prev_next_buttons_location': 'bottom',
    'style_external_links': True,
}
latex_engine = 'xelatex'
```

4. To generate the notes in pdf format, use `make latex`, which generates lots of `tex` files in `./build/latex`. Switch to `build/latex` and run `make`. Assume that you have installed the software to compile `tex` files. It will generate `notes.pdf`.



This page describes commonly used git commands.

## 2.1 Commands

### 2.1.1 rev-parse

It is quite common to get the root directory of the repository with the command:

```
git rev-parse --show-toplevel
```

For instance, the above command executed in this repository prints something like as follows:

```
/xxx/notes
```

The following shows its usage in a Python script:

```
#!/usr/bin/env python3

import subprocess

d = (
    subprocess.check_output(["git", "rev-parse", "--show-toplevel"])
    .decode("ascii")
    .strip() # remove the trailing \n
)
print(d) # /path/to/notes
```

It can also be used in bash script:

```
root_dir=$(git rev-parse --show-toplevel)
echo "root_dir ${root_dir}"
```

help git-rev-parse outputs helpful information for git rev-parse. In particular, it explains the differences among HEAD~, HEAD~n, HEAD^, and HEAD^n. The following shows the help information about it:

```
<rev>^[<n>], e.g. HEAD^, v1.5.1^0
  A suffix ^ to a revision parameter means the first parent of that commit object. ^
  ↪<n> means the <n>th parent
```

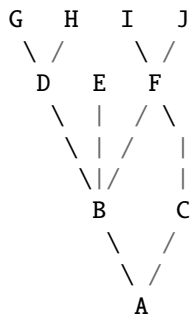
(continues on next page)

(continued from previous page)

(i.e. `<rev>^` is equivalent to `<rev>^1`). As a special rule, `<rev>^0` means the commit itself and is used when `<rev>` is the object name of a tag object that refers to a commit object.

`<rev>~[<n>]`, e.g. `HEAD~`, `master~3`

A suffix `~` to a revision parameter means the first parent of that commit object. A suffix `~<n>` to a revision parameter means the commit object that is the `<n>`th generation ancestor of the named commit object, following only the first parents. I.e. `<rev>~3` is equivalent to `<rev>^^^` which is equivalent to `<rev>^1^1^1`. See below for an illustration of the usage of this form.



$A = A^0$   
 $B = A^1 = A^1_1 = A_{~1}$   
 $C = A^2$   
 $D = A^{11} = A^{1^1}_1 = A_{~2}$   
 $E = B^2 = A^{12}$   
 $F = B^3 = A^{13}$   
 $G = A^{111} = A^{1^1^1}_1 = A_{~3}$   
 $H = D^2 = B^{12} = A^{112} = A_{~2}^2$   
 $I = F^1 = B^{13} = A^{113}$   
 $J = F^2 = B^{132} = A^{1132}$

## 3.1 Installation

### 3.1.1 macos

Refer to <https://docs.docker.com/desktop/mac/install/>.



## **4.1 TikZ**

### **4.1.1 Basics**



This page describes commonly used git commands.

## 5.1 Decoding

```
CompactLattice compact_lat;  
decoder.GetLattice(true, &compact_lat);  
  
CompactLattice compact_best_path;  
CompactLatticeShortestPath(compact_lat, &compact_best_path);  
  
Lattice best_path;  
ConvertLattice(compact_best_path, best_path);  
  
std::vector<int32_t> tokens;  
std::vector<int32_t> words;  
LatticeWeight weight;  
GetLinearSymbolSequence(best_path, &tokens, &words, &weight);
```

- decoder/simple-decoder.{h,cc}





## 6.1 asyncio

### 6.1.1 Hello World

### 6.1.2 References

- PEP 234 – Iterators  
<https://peps.python.org/pep-0234/>
- Why does defining `__getitem__` on a class make it iterable in python?  
<https://localcoder.org/why-does-defining-getitem-on-a-class-make-it-iterable-in-python>
- PEP 255 – Simple Generators  
<https://peps.python.org/pep-0255/>
- Curious Course on Coroutines and Concurrency  
[https://www.youtube.com/watch?v=Z\\_OAlhXziw&ab\\_channel=DavidBeazley](https://www.youtube.com/watch?v=Z_OAlhXziw&ab_channel=DavidBeazley)  
By David Beazley.
- Generator Tricks for Systems Programmers  
<https://www.dabeaz.com/generators2/>
- Generators: The Final Frontier  
<[https://www.youtube.com/watch?v=5-qadlG7tWo&ab\\_channel=DavidBeazley](https://www.youtube.com/watch?v=5-qadlG7tWo&ab_channel=DavidBeazley)>  
By David Beazley.

## 6.2 argv

From the doc <https://docs.python.org/3/library/sys.html>:

The `list` of command line arguments passed to a Python script. `argv[0]` **is** the script name (it **is** operating system dependent whether this **is** a full pathname **or not**). If the command was executed using the `-c` command line option to the interpreter, `argv[0]` **is set** to the string `'-c'`. If no script name was passed to the Python interpreter, `argv[0]` **is** the empty string.

Note that `argv` is at least of size 1, though `argv[0]` may be an empty string.

```
import sys
print(sys.argv)
```

## 7.1 sort

Sort files in the folder `t`. The filename has the pattern `xxx.n.txt`, where `n` is some numerical value. Also, exclude `xxx.100.txt`.

```
find ./t -name "xxx*.txt" ! -name "xxx.100.txt" -print0 | sort -z -t. -k2 -n | xargs -r0
```

## 7.2 echo

Generate a binary file:

```
echo -n -e '\x30\x31\x32' > a.bin  
hexdump a.bin
```



## CUDA

### 8.1 Installation

#### 8.1.1 CUDA 10.1.243

```
./cuda_10.1.243_418.87.00_linux.run --silent --toolkit --installpath=/ceph-data4/fangjun/  
↪software/cuda-10.1.243 --no-opengl-libs --no-drm --no-man-page  
  
# Install cuDNN  
cd /ceph-data4/fangjun/software/cuda-10.1.243  
tar xvf /ceph-sh0/fangjun/cudnn/cudnn-10.1-linux-x64-v8.0.4.30.tgz --strip-components=1
```

```
#!/usr/bin/env bash  
  
export CUDA_HOME=/ceph-data4/fangjun/software/cuda-10.1.243  
export PATH=$CUDA_HOME/bin:$PATH  
export LD_LIBRARY_PATH=$CUDA_HOME/lib64:$LD_LIBRARY_PATH  
  
# See /ceph-fj/fangjun/py38/lib/python3.8/site-packages/torch/share/cmake/Caffe2/Modules_  
↪CUDA_fix/upstream/FindCUDA.cmake  
export CUDA_TOOLKIT_ROOT_DIR=$CUDA_HOME  
export CUDA_TOOLKIT_ROOT=$CUDA_HOME  
export CUDA_BIN_PATH=$CUDA_HOME  
export CUDA_PATH=$CUDA_HOME  
export CUDA_INC_PATH=$CUDA_HOME/targets/x86_64-linux
```

#### 8.1.2 CUDA 11.0.3

```
./cuda_11.0.3_450.51.06_linux.run --silent --toolkit --installpath=/ceph-data4/fangjun/  
↪software/cuda-11.0.3 --no-opengl-libs --no-drm --no-man-page  
  
# Install cuDNN  
cd /ceph-data4/fangjun/software/cuda-11.0.3  
tar xvf /ceph-sh0/fangjun/cudnn/cudnn-11.0-linux-x64-v8.0.4.30.tgz --strip-components=1
```

### 8.1.3 CUDA 11.3.1

```
./cuda_11.3.1_465.19.01_linux.run --silent --toolkit --installpath=/ceph-data4/fangjun/  
↪software/cuda-11.3.1 --no-opengl-libs --no-drm --no-man-page  
cd /ceph-data4/fangjun/software/cuda-11.3.1  
tar xvf /ceph-sh0/fangjun/cudnn/cudnn-11.3-linux-x64-v8.2.1.32.tgz --strip-components=1
```

### 8.1.4 CUDA 11.5.2

```
./cuda_11.5.2_495.29.05_linux.run --silent --toolkit --installpath=/ceph-data4/fangjun/  
↪software/cuda-11.5.2 --no-opengl-libs --no-drm --no-man-page  
cd /ceph-data4/fangjun/software/cuda-11.5.2  
tar xvf /ceph-sh0/fangjun/cudnn/cudnn-linux-x86_64-8.3.2.44_cuda11.5-archive.tar.xz --  
↪strip-components=1
```

### 8.1.5 CUDA 11.6.1

```
./cuda_11.6.1_510.47.03_linux.run --silent --toolkit --installpath=/ceph-data4/fangjun/  
↪software/cuda-11.6.1 --no-opengl-libs --no-drm --no-man-page  
cd /ceph-data4/fangjun/software/cuda-11.6.1  
tar xvf /ceph-sh0/fangjun/cudnn/cudnn-11.3-linux-x64-v8.2.1.32.tgz --strip-components=1
```

This page describes commonly used git commands.

## **9.1 DDP**

### **9.1.1 Initialization**





## 10.1 Install

### 10.1.1 formatter

Install <https://github.com/google/google-java-format>

```
wget https://github.com/google/google-java-format/releases/download/v1.15.0/google-java-format-1.15.0-all-deps.jar
```

Create a script with filename `google-java-format`:

```
#!/usr/bin/env bash

java -jar /ceph-sh0/fangjun/download/google-java-format-1.15.0-all-deps.jar $@
```

`chmod +x google-java-format` and add the path to `PATH`.

### 10.1.2 JDK

Go to <https://www.oracle.com/java/technologies/downloads/#java17> and download

```
wget https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz
mkdir /ceph-fj/fangjun/software/
tar xvf jdk-17_linux-x64_bin.tar.gz -C /ceph-fj/fangjun/software
```

And then set the following environment variables:

```
export JAVA_HOME=/ceph-fj/fangjun/software/jdk-17.0.3
export PATH=$JAVA_HOME/bin:$JAVA_HOME
```

## 10.2 Hello world

Listing 1: Hello.java

```
// Usage 1:
//  java Hello.java
// Usage 2:
//  javac Hello.java
//  java Hello
//
// Note:
//  - "javac Hello.java" generates a file "Hello.class"
//  - "java Hello" takes as input "Hello.class" and executes it
//
class Hello {
    public static void main(String[] args) {
        System.out.println("hello world");
    }
} // There is no ';' here
```

Listing 2: EqualTest.java

```
class EqualTest {
    public int i;

    public EqualTest(int a) {
        this.i = a;
    }

    public boolean equals(Object anObject) {
        if (this == anObject) {
            return true;
        }
        if (anObject instanceof EqualTest) {
            return this.i == ((EqualTest) anObject).i;
        }
        return false;
    }

    public static void main(String[] args) {
        EqualTest e1 = new EqualTest(10);
        EqualTest e2 = new EqualTest(10);

        System.out.println(e1 == e2); // false, compare the reference
        System.out.println(e1 != e2); // true
        System.out.println(e1.equals(e2)); // true, compare the contained value
    }
}
```

## 10.3 Reference

- <https://docs.oracle.com/javase/tutorial/>
- <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>
- <https://github.com/openjdk/jdk.git>

Clone it and you can find the source code in `src/java.base/share/classes/java/lang/System.java` for `java.lang.System`.



## JAVASCRIPT

### 11.1 Hello world

```
console.log('hello world')  
console.log(eval('3 + 5'))
```

To write multi-line javascript, use shift + Enter for a new line.

```
(function(){  
    "use strict";  
    /* Start of your code */  
    function greetMe(yourName) {  
        alert('Hello ' + yourName);  
    }  
  
    greetMe('World');  
    /* End of your code */  
})();
```

It is case sensitive. Statements are separated by ;. Comments are the same as in C/C++.

### 11.2 TODOs

1. This page [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/Installing\\_basic\\_software](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Installing_basic_software) lists some tools to minify code:
  - WebPack: <https://webpack.js.org/>
  - Grunt: <https://gruntjs.com/>
  - Gulp: <https://gulpjs.com/>
2. Color picker tool: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Colors/Color\\_picker\\_tool](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool)
3. Google font: <https://fonts.google.com/> and [https://developers.google.com/fonts/docs/getting\\_started](https://developers.google.com/fonts/docs/getting_started)



## 12.1 Hello world

Listing 1: hello\_world.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

### 12.1.1 comments

```
<!-- this is a comment -->
```

### 12.1.2 images

```
</img>
</img>
</img>
```

### 12.1.3 ordered lists

```
<p> The following points </p>

<ol>
  <li> First </li>
  <li> Second </li>
</ol>
```

### 12.1.4 unordered lists

```
<p> The following points </p>

<ul>
  <li> foo </li>
  <li> bar </li>
</ul>
```

### 12.1.5 links

```
<a href="https://www.google.com">some text</a>
```

## 12.2 References

- Structuring the web with HTML  
<https://developer.mozilla.org/en-US/docs/Learn/HTML>



## 13.1 Hello world

### 13.1.1 comment

```
/* this is a comment */
```

```
p { color: red; }
```

Then, in some html file, use:

```
<link href="abc/foo.css" rel="stylesheet">
```

### 13.1.2 Selector

- tag name or element name: e.g., `p` selects `<p>`; `h1` selects `<h1>`.
- ID: e.g., `#my-id` selects `<a id="my-id">` or `<p id="my-id">`
- class: e.g., `.my-class` selects `<a class="my-class">` and `<p class="my-class">`
- attribute: e.g., `img[src]` selects `` but not `<img>`

See [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics#different\\_types\\_of\\_selectors](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics#different_types_of_selectors) and [https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors) for more.

Example with multiple rules:

Example with multiple selectors:

## 13.2 References

- CSS basics

[https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics)