

# GSoC 2017 Proposal to Kivy (Python Software Foundation)

## Project: KV Compiler: A compiler for the KV language

### Sub-organization information

Kivy

### Mentors

Matthew Einhorn (matham)

### Student Information

Name: Yash Jain

Email: [yashjain.lnm@gmail.com](mailto:yashjain.lnm@gmail.com)

Telephone: +919414419320 / +919660243960

Time Zone: Jaipur, India UTC+5:30

IRC: [yaki29@irc.freenode.net](irc://yaki29@irc.freenode.net)

Source Control Username: <http://www.github.com/yaki29>

Facebook: <https://www.facebook.com/profile.php?id=100003284453761>

Blogs: <https://yaki29.github.io/Blog> (Will be available)

### University Information

University: The LNM Institute of Information Technology, Jaipur

Major: Computer Science and Engineering

Current Year: 2nd Year

Expected Graduation Completion: In June 2019

Degree: B-Tech

### Project Proposal Information

**Proposal Title:** **KV Compiler: A compiler for the KV language**

**Proposal Abstract:** This Proposal is based on understanding of kv language in the [ideas page](#) provided by Kivy Organization. The goal of the project will be to create a compiler which compiles kv code into python code, with debug/optimization option.

#### **Project Description:**

A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language). It might include debugging and optimization as well. Here in this case the source language would be kv-lang and the target language would be python.

Currently, the bindings are not at all optimized because upon each widget creation all of the rules are re-evaluated and bound. This process can be significantly optimized by pre-compiling the kv code, especially the bindings. Hence we need a better and optimized compiler.

- **Access to required hardware:**

I personally own Linux, Windows and Android. If anything else is needed I will buy it within a week.

- **Dividing the work flow:**

My summer vacations will start from mid-May and will be ending in mid-July, So I have planned my workflow accordingly. I will finish the basic understanding(theory) part as soon as possible, understand the alpha stage compiler and jump to the next part as explained in the timeline. I will be dividing my work in 4 parts.

- Phase-1: Understand the alpha stage compiler. (Explained in timeline).
- Phase-2: Add binding features and basic optimization methods. (Explained in timeline)
- Phase-3: Consistent and Optimize compiler is hoped, with testing the code. (Explained in timeline)
- Phase-4: Winding up the project(extensive testing and complex optimization). (Explained in timeline)

## Timeline:

<p><i>Up to 23<sup>rd</sup> May</i></p>	<p><i>For compiler working to understand I'll have to study a lot about kivy-bindings, widget-behaviors, how they work, how they show changes, when they are changed etc.</i></p> <p><i>I will be reading regularly as suggested by my mentors. I would gain as much knowledge as I can regarding optimization and debugging.</i></p> <p><i>I will be in touch with my mentors and take suggestions.</i></p> <p><i>During this time, if my mentor thinks that I am ready to start programming then I will start working on my Phase-1.</i></p>
<p><b>23<sup>rd</sup> May – 17<sup>th</sup> June</b></p>	<p style="text-align: center;"><b><u>Phase-1</u> (Week 1 - 4)</b></p> <ul style="list-style-type: none"> <li>• During this phase I will be working on alpha stage compiler, understand how it works, flow of control and how it is different from the existing one.</li> <li>• Enlist missing parts of the compiler and the features to be added.</li> <li>• Present optimization techniques, keeping consistency as my first motive.</li> <li>• Documentation and examples will be done along with this.</li> </ul>

23 <sup>rd</sup> May – 5 <sup>th</sup> June	<p><b>(Week 1 &amp; 2)</b></p> <ul style="list-style-type: none"> <li>• Start writing a basic compiler which at least compiles basic examples of kv-language.</li> <li>• Perform some testing and debug the code and keep on adding features as reviewed by the mentors.</li> </ul>
6 <sup>th</sup> June – 17 <sup>th</sup> June	<p><b>(Week 3 &amp; 4)</b></p> <ul style="list-style-type: none"> <li>• Work on batching the bindings, kivy-behaviors and clear up where currently kv is ill defined.</li> </ul>
18 <sup>th</sup> June – 7 <sup>th</sup> July	<p><b>Phase-2 (Week 5 &amp; 6 &amp; 7)</b></p> <ul style="list-style-type: none"> <li>• Add features to the compiler which makes parse tree and rebind the bindings when they are changed or new ones are added.</li> <li>• Documentation and examples will be done along with this.</li> </ul>
18 <sup>th</sup> June – 26 <sup>th</sup> June	<p><b>(Week-5)</b></p> <ul style="list-style-type: none"> <li>• Code the compiler that does batching of bindings and rebind some of them when they are changed.</li> </ul>
Mid-Term Evaluation	<p><b>(6<sup>th</sup> Week)</b></p> <ul style="list-style-type: none"> <li>• Make Preparation for the mid-term evaluation.</li> <li>• Seek feedback and make revisions based on that.</li> <li>• Submit the evaluations before 30<sup>th</sup> June.</li> </ul>
28 <sup>th</sup> June – 7 <sup>th</sup> July	<p><b>(Week 6 &amp; 7)</b></p> <ul style="list-style-type: none"> <li>• Will take the feedback from the evaluation and make changes (if required).</li> <li>• Continue with the previous task. If done, then move to next step.</li> <li>• If completed before time, then jump to Phase-3</li> </ul>
07 <sup>th</sup> July – 24 <sup>th</sup> July	<p><b><u>Phase-3</u> Week (8, 9 &amp; 10)</b></p> <ul style="list-style-type: none"> <li>• During this period, I will be working more on Consistency and Optimization methods and techniques.</li> <li>• Documentation and examples will be done along with it.</li> </ul>
07 <sup>th</sup> July – 24 <sup>th</sup> July	<p><b>Week (8 &amp; 9 &amp; 10)</b></p> <ul style="list-style-type: none"> <li>• Discuss with the mentors about various methods and techniques that can be used to optimize the compiler.</li> <li>• Apply those techniques and perform regular testing and debug the code along with it.</li> <li>• Submit the evaluations before 28<sup>th</sup> July.</li> </ul>

1 <sup>st</sup> August – 23 <sup>rd</sup> August	<p style="text-align: center;"><b><u>Phase-4 Week (11 &amp; 12 &amp; 13)</u></b></p> <ul style="list-style-type: none"> <li>• During this phase, I will be working on complex features.</li> <li>• Documentation and examples will be done along with it.</li> </ul>
1 <sup>st</sup> August – 18 <sup>th</sup> August	<ul style="list-style-type: none"> <li>• Complex Optimization techniques will be implemented, as discussed with the mentors.</li> <li>• Extensive testing will be my main focus during this phase.</li> </ul>
19 <sup>th</sup> August -23 <sup>rd</sup> August	<ul style="list-style-type: none"> <li>• Continue to Implement features (if left).</li> <li>• Complete any missing documentations.</li> <li>• Complete evaluations and send them before 29<sup>th</sup> August.</li> </ul>
24 <sup>th</sup> August – 29 <sup>th</sup> August	<ul style="list-style-type: none"> <li>• A Buffer of one week has been kept for any unpredictable delay.</li> </ul>
Onwards	<ul style="list-style-type: none"> <li>• Keep contributing to Kivy and its sister projects and make use of these features.</li> </ul>

**Link to a patch/code sample, preferably one you have submitted to your sub-org (\*):**  
**I have implemented the following:**

1. [Facelock](#) using Computer Vision techniques. (Merged)
2. [BeautifulSoup4 recipe](#) for android. (Waiting for approval)
3. [Camera example](#) for kivy. (Merged)
4. Corrected Basic [Documentation mistakes](#) in p4a.

## Other Commitments:

- ◆ Have you applied to any other organization? **No.**
- ◆ Do you have any other commitments during the main GSoC time period? **No.**
- ◆ Do you have exams or classes that overlap with this period? **No.**  
(Even if there is some kind of emergency, in that case I will work extra hard during the weekends.)

## Why am I apt. for this project:

Familiarity with Kivy and its coding style. I have an extreme interest and curiosity towards coding and have been doing it for past 2 years with renowned language Python, Java and C. I have been contributing to Kivy and its sister projects for past 5-6 months(mainly [Python-for-android](#) and [Kivy-Garden](#)). I am working on a [2-pass Assembler](#) at my college and some python projects in Kivy, some of them could be found on my GitHub profile.

## Proposal Improvements:

- It will definitely consist of an optimization/debug option, one more option can be added further which recursively compiles all the files in the working directory, so that user doesn't have to sit and compile all the files of his kivy-project.
- What if user just wants to compile some of the custom kv files, he/she can just enter the file names using a space between them it will identify the multiple files and compile them one by one. That will make the compiler more handy.