
Python Core Sprint 2020

Mariatta

Oct 23, 2020

CONTENTS:

1	Code of Conduct	3
2	Schedule	5
3	Participant List	7
4	Sprint Projects	9
4.1	Sprint Blog Post	9
4.2	C-API/runtime/subinterpreters	9
4.3	email	10
4.4	stdlib modules	10
4.5	Parser improvements	10
4.6	Core workflow	10
4.7	General issue triage/ PR reviews	11
4.8	Release management	11
4.9	Typing	11
4.10	asyncio	12
4.11	GC improvements	12
4.12	Optimize startup time	12
4.13	Buildbot Configuration	12
4.14	Deprecate old modules	12
4.15	F-strings	13
4.16	Dataclasses	13
4.17	Language Design Discussions	13
4.18	Windows	13
4.19	MacOS	13
4.20	DSLs and support for dynamic and/or annotated ASTs	14
4.21	Performance improvements	14
4.22	Increasing new contributors; mentorship	14
4.23	importlib.resources	14
4.24	multiphase init and heap type	15
4.25	Documentation	15
4.26	PEP 447	15
4.27	Security / SSL	15
4.28	Gilectomy	15
4.29	New PEP	16
4.30	Posix subprocess	16
4.31	Build System	16
4.32	PEP 622/634 Structural Pattern Matching	16
4.33	Wheel interpreter naming for 3.10	16

4.34	Hypothesis standard library tests	16
4.35	Auto-Formatting the CPython Codebase	16
5	Communication	19
5.1	Starting a new meeting	19
5.2	Note taking	19
6	Discord Setup Guide	21
6.1	Privacy settings	21
6.2	Audio settings	22
6.3	UI settings	23
6.4	Joining Python Discord	24
6.5	Channel structure & navigation	26
6.6	Renaming categories & creating new channels	30
7	Python community Q&A session	33
7.1	Core devs participating in the Community Q&A	33
8	Social Activities	35
8.1	Escape Room	35
8.2	Online Board Game	36
8.3	Unstructured social time	37
9	Tech check session/tech day	39

Issue Tracker and Discussions: <https://github.com/python/core-sprint>

For questions and suggestions about the sprint, please open an issue or a discussion item in the [core-sprint](#) repo.

For sensitive and private discussions, please get in touch with one of the organizers: Mariatta, Kyle Stanley or Ewa Jodlowska.

The sprint is scheduled for **Oct 19 - 23, 2020**. This will be an online event instead of the in-person sprint.

The sprint is normally a weeklong (5 work days) sprint, where we would have focus time to work on CPython, finish your PEPs, review pull requests, etc.

All core devs, Python triage members, and active core dev mentees are invited to participate.

Sign up form: <https://forms.gle/84NkyYt5g616fs1r5>

After signing up through the form, please add yourself to this doc by creating a pull request. You'll want to add yourself to *Participant List*, *Sprint Projects*, and if applicable *Python community Q&A session*.

Help improve this documentation by opening an issue or creating a pull request at <https://github.com/python/core-sprint>

CODE OF CONDUCT

The PSF's [Code of Conduct](#) applies and will be enforced.

If you want to report incidents that occur during the sprint, please report it to The PSF's Code of Conduct working group. They can be reached by emailing conduct-wg@python.org

Among the core sprint attendees, Carol Willing and Brett Cannon are members of the Code of Conduct work group.

SCHEDULE

The sprint will take place from **Oct 19 to 23, 2020**.

We will have a live kick-off meeting on **Monday Oct 19, 9 AM PDT (4 PM UTC)**, and a send-off meeting on **Friday Oct 23, 9 AM PDT (4 PM UTC)**.

The meetings will be recorded so those unable to attend the meeting can watch it afterwards.

Other planned activities:

- *Python community Q&A session*: 30 minutes Q & A Session
- Core devs & Steering council Q&A session
- *Social Activities*
- *Tech check session/tech day* session: the day before the sprint, for anyone who needs help with setting up Zoom, Discord, etc.
- Group Photo

Shared Google Calendar [Python Virtual Core Sprint Calendar](#)

Date	Time	Activity
Sunday, Oct 18, 2020	3 PM - midnight Eastern	Tech check session
Monday, Oct 19, 2020	9 AM Pacific (UTC-7)	Kick-off meeting (live, and recorded), followed with group photo
Tuesday, Oct 20, 2020	3 PM Central	<i>Python community Q&A session</i>
Wednesday, Oct 21, 2020	9 AM Pacific	Core devs & Steering council Q&A session
Friday, Oct 22, 2020	9 AM Pacific	Send-off meeting (live, and recorded), followed with another group photo

PARTICIPANT LIST

The following is a list of core developers, bug triage, and mentees participating in Python Core Sprint 2020.

See also:

Sprint Projects for the list of sprint projects.

All core devs, Python triage members, and active core dev mentees are invited to participate.

You are a core developer if:

- You're listed as a member of [Python core](#) team on GitHub, or
- You're listed in Devguide's [Developer Log](#)

You are a Python triage members if:

- You have the “Developer” role in [bugs.python.org](#), or
- You're listed as a member of [Python triage](#) team on GitHub

If you have not signed up and would like to participate, please fill in this form: <https://forms.gle/84NkyYt5g616fs1r5>

After signing up through the form, please add yourself to this doc by creating a pull request. You'll want to add yourself to *Participant List*, *Sprint Projects*, and if applicable *Python community Q&A session*.

See also:

[Timezone converter](#).

Name	Status	Timezone	GitHub Handle	Notes
Senthil Kumaran	Core Developer	UTC+9	orsenthil	
Dong-hee Na	Core Developer	UTC+9	corona10	
Inada Naoki	Core Developer	UTC+9	methane	
Hai Shi	Mentee	UTC+8	shihai1991	Mentored by Victor Stinner
Batuhan Taşkaya	Triage	UTC+3	Isidental	
Lysandros Nikolaou	Core Developer	UTC+3	lysnikolaou	
Łukasz Langa	Core Developer	UTC+2	ambv	
Victor Stinner	Core Developer	UTC+2	vstinner	
Tal Einat	Core Developer	UTC+2	taleinat	
Petr Viktorin	Core Developer	UTC+2	encukou	
Stefan Behnel	Core Developer	UTC+2	scoder	
Ronald Oussoren	Core Developer	UTC+2	ronaldoussoren	
Thomas Wouters	Core Developer	UTC+2	Yhg1s	
Christian Heimes	Core Developer	UTC+2	tiran	
Pablo Galindo Salgado	Core Developer	UTC+1	pablogsal	
Vinay Sajip	Core Developer	UTC+1	vsajip	

continues on next page

Table 1 – continued from previous page

Lewis Gaul	Mentee	UTC+1	LewisGaul	Mentored by Eric Snow (excited too!)
Mark Shannon	Core Developer	UTC+1	markshannon	
Filipe Laíns	Mentee	UTC+1	FFY00	Mentored by Jason R. Coombs
Irit Katriel	Triage	UTC+1	iritkatriel	
Eric V. Smith	Core Developer	UTC-4	ericvsmith	
Ammar Askar	Triage	UTC-4	ammaraskar	
Ned Deily	Core Developer	UTC-4	ned-deily	
Jason R. Coombs	Core Developer	UTC-4	jaraco	
Kyle	Core Developer	UTC-4	aeros	
Joannah Nanjeyye	Core Developer	UTC-4	nanjeyyejoannah	
Terry Jan Reedy	Core Developer	UTC-4	terryjreedy	
Paul Ganssle	Core Developer	UTC-4	pganssle	
Éric Araujo	Core Developer	UTC-4	merwok	
Fred Drake	Core Developer	UTC-4	freddrake	
Andrew Kuchling	Core Developer	UTC-5	akuchling	
Zach Ware	Core Developer	UTC-5	zware	
Emily Morehouse	Core Developer	UTC-6	emilyemorehouse	
Eric Snow	Core Developer	UTC-6	ericssnowcurrently	so excited!
Barry Warsaw	Core Developer	UTC-7	warsaw	
Mariatta	Core Developer	UTC-7	mariatta	
Guido van Rossum	Core Developer	UTC-7	gvanrossum	
Carol Willing	Core Developer	UTC-7	willingc	
Neil Schemenauer	Core Developer	UTC-7	nascheme	
Brett Cannon	Core Developer	UTC-7	brettcannon	
Yury Selivanov	Core Developer	UTC-7	1st1	
Gregory P. Smith	Core Developer	UTC-7	gps	I'm open to time-shifting a bit.
Larry Hastings	Core Developer	UTC-7	larryhastings	
Brandt Bucher	Core Developer	UTC-7	brandtbucher	
Trent Nelson	Core Developer	UTC-8	tpn	
Steve Dower	Core Developer	UTC-8	zooba	

SPRINT PROJECTS

See also:

[Participant List](#) for the list of all sprint participants.

As it gets closer to the sprint, we will ensure that all participants will have access to all the notes linked below.

4.1 Sprint Blog Post

Please help us write the sprint summary blog post! This will be posted in The PSF's blog. It would help us if you could share your daily updates below.

[Draft of the Blog Post](#)

4.2 C-API/runtime/subinterpreters

- Joannah Nanjekye
- Pablo Galindo Salgado
- Neil Schemenauer
- Petr Viktorin
- Stefan Behnel
- Trent Nelson
- Victor Stinner
- Dong-hee Na
- Christian Heimes
- Eric Snow
- Hai Shi
- Gregory P. Smith
- Lewis Gaul
- Barry Warsaw

[C-API/subinterpreters Notes](#).

4.3 email

- Barry Warsaw

[email Notes](#).

4.4 stdlib modules

- Barry Warsaw
- Batuhan Taşkaya
- Pablo Galindo Salgado
- Senthil Kumaran
- Stefan Behnel
- Dong-hee Na
- Guido van Rossum
- Vinay Sajip
- Éric Araujo
- Irit Katriel

[stdlib modules Notes](#).

4.5 Parser improvements

- Pablo Galindo Salgado
- Lysandros Nikolaou
- Guido van Rossum

[Parser improvements Notes](#).

4.6 Core workflow

- Mariatta: [miss-islington](#), [bedevere issue #265](#)
- Brett Cannon: GitHub action to calculate SC voter roll, VS Code dev container, GitHub action to detect when `importlib.h` is out-of-date
- Tal Einat: Project status dashboard

[Core workflow Notes](#).

4.7 General issue triage/ PR reviews

- Mariatta
- Petr Viktorin
- Andrew Kuchling
- Zach Ware
- Lysandros Nikolaou
- Batuhan Taşkaya
- Steve Dower
- Guido van Rossum
- Vinay Sajip
- Gregory P. Smith
- Brandt Bucher
- Mark Shannon
- Inada Naoki
- Carol Willing
- Éric Araujo
- Irit Katriel

General issue triage/ PR reviews Notes.

4.8 Release management

- Łukasz Langa
- Ned Deily
- Pablo Galindo Salgado

Release management Notes.

4.9 Typing

- Łukasz Langa
- Guido van Rossum

Typing Notes.

4.10 asyncio

- Carol Willing
- Kyle
- Yury Selivanov
- Guido van Rossum

[asyncio Notes](#).

4.11 GC improvements

- Neil Schemenauer
- Pablo Galindo Salgado
- Joannah Nanjekye
- Mark Shannon

[GC improvements Notes](#).

4.12 Optimize startup time

- Neil Schemenauer
- Eric Snow
- Mark Shannon
- Barry Warsaw

[Optimize startup time Notes](#).

4.13 Buildbot Configuration

- Zach Ware

[Buildbot Configuration Notes](#).

4.14 Deprecate old modules

- Brett Cannon: Pkgutil

[Deprecate old modules Notes](#).

4.15 F-strings

- Eric V. Smith

[F-strings Notes.](#)

4.16 Dataclasses

- Eric V. Smith

[Dataclasses Notes.](#)

4.17 Language Design Discussions

- Stefan Behnel
- Pablo Galindo Salgado
- Emily Morehouse
- Batuhan Taşkaya
- Thomas Wouters
- Guido van Rossum
- Eric Snow
- Vinay Sajip
- Barry Warsaw

[Language Design Discussions Notes.](#)

4.18 Windows

- Trent Nelson

[Windows Notes.](#)

4.19 MacOS

- Ned Deily
- Ronald Oussoren

[MacOS Notes.](#)

4.20 DSLs and support for dynamic and/or annotated ASTs

- Emily Morehouse
- Guido van Rossum

DSLs and support for dynamic and/or annotated ASTs Notes.

4.21 Performance improvements

- Tal Einat

Performance improvements Notes.

4.22 Increasing new contributors; mentorship

- Tal Einat
- Guido van Rossum
- Eric Snow
- Joannah Nanjekye
- Mariatta
- Carol Willing
- Ammar Askar
- Batuhan Taşkaya
- Kyle Stanley
- Lewis Gaul
- Zach Ware

Increasing new contributors; mentorship Notes.

4.23 importlib.resources

- Jason R. Coombs
- Filipe Laíns
- Barry Warsaw

importlib.resources Notes.

4.24 multiphase init and heap type

- Hai Shi
- Dong-hee Na
- Eric Snow

[multiphase init and heap type Notes.](#)

4.25 Documentation

- Terry Jan Reedy
- Vinay Sajip
- Gregory P. Smith
- Mariatta
- Carol Willing
- Fred Drake

[Documentation Notes.](#)

4.26 PEP 447

- Ronald Oussoren

[PEP 447 Notes.](#)

4.27 Security / SSL

- Christian Heimes

[Security / SSL Notes.](#)

4.28 Gilectomy

- Larry Hastings

4.29 New PEP

- Larry Hastings

4.30 Posix subprocess

- Gregory P. Smith

4.31 Build System

- Gregory P. Smith

4.32 PEP 622/634 Structural Pattern Matching

- Brandt Bucher
- Barry Warsaw
- Carol Willing

4.33 Wheel interpreter naming for 3.10

Or: Python 3.10 and the case of that pesky double-digit minor version number ([PR](#))

- Brett Cannon
- Ned Deily
- Pablo Galindo Salgado
- Barry Warsaw

4.34 Hypothesis standard library tests

Trying to follow up on the proposal by Zac Hatfield-Dobbs from the language summit this year.

- Paul Ganssle

4.35 Auto-Formatting the CPython Codebase

- Ammar Askar
- Batuhan Taşkaya
- Dong-hee Na
- Gregory P. Smith
- Pablo Galindo Salgado

- Tal Einat

COMMUNICATION

The sprint is attended by people from various timezones, so expect more asynchronous discussions throughout the sprint.

We will be using Discord to communicate during the sprint. See the [Discord Setup Guide](#) for more details.

Additionally we will use Zoom for the kick-off and send-off meetings and for group video chats.

5.1 Starting a new meeting

You're free to schedule a meeting with another participant anytime. When you scheduled a meeting, please add the meeting to the shared Google Calendar. Once you add the meeting to Google Calendar, it will be automatically announced on Discord. Other sprint attendees will be able to discover your scheduled meeting, and join in if they're interested too.

Remember to take notes during your meeting!

5.2 Note taking

We heard that people didn't like HackMD from last year's sprint. So this time we have decided to use Google Docs. We will be creating the Google Docs and share the links.

When you're in a group meeting, please remember to take down notes, so those unable to attend the meeting can get up speed.

We have created a Google Doc for each project. You can find the link to the notes in [Sprint Projects](#). We will ensure that each participant will have access to all the docs as it gets closer to the sprint.

DISCORD SETUP GUIDE

Welcome to the Discord setup guide! The information in this guide is mostly specific to sprint participants, and will cover:

- *Privacy settings*
- *Audio settings*
- *UI settings*
- *Joining Python Discord*
- *Channel structure & navigation*
- *Renaming categories & creating new channels*

For additional details that cover the fundamentals of using Discord in general, I would recommend their [Beginner's Guide](#). I'll go over some basics, but the above guide is more detailed for first-time users.

For those who are very familiar with using Discord, feel free to skip to “Joining Python Discord”.

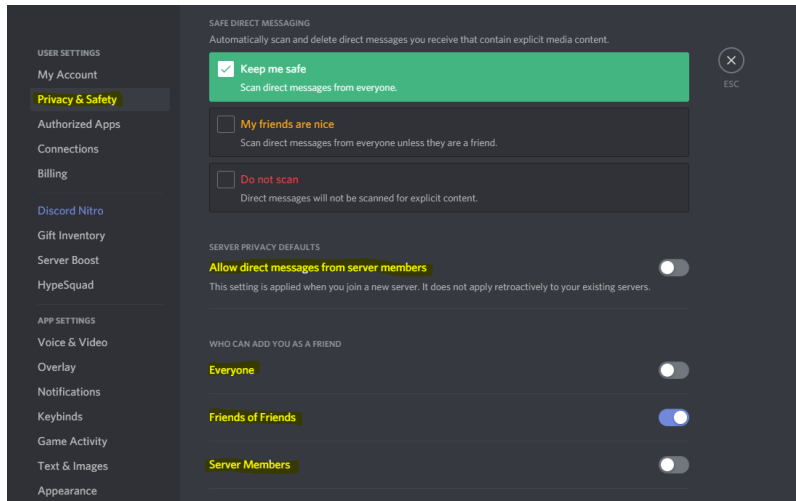
As a very general Discord recommendation, I would advise downloading the desktop client at <https://discord.com/download>. Generally, it looks and functions better as a full-screen desktop application rather than in the browser as a web application, but either works reasonably well.

If anything is unclear, either within this guide or in using Discord in general, feel free to reach out to the author directly at aeros167@gmail.com or via DM on discuss.python.org to aeros.

6.1 Privacy settings

Prior to joining any public server (especially highly populated ones), it is recommended to configure account privacy settings to disallow DMs from server members, and strictly allow friend requests from existing friends.

After clicking the gear box in the lower left corner, proceed to “User Settings > Privacy & Safety”:



In the above image, the relevant options are highlighted:

“Allow direct messages from server members”: **Off**. Without this setting, any member on a mutual server would be able to directly send messages.

“Who can add you as a friend?”: **Friends of Friends only**. Using this setting, only those who are friends with your existing friends will be able to send friend requests. This requires at least one mutual friend in order for others to add you, but it will drastically reduce the pool of potential users that can send friend requests.

As for that one mutual friend, **I would recommend for participants to send a friend request to aeros#0717**, and then add each other. This would allow each participant of the sprint to easily DM anyone they’ve added.

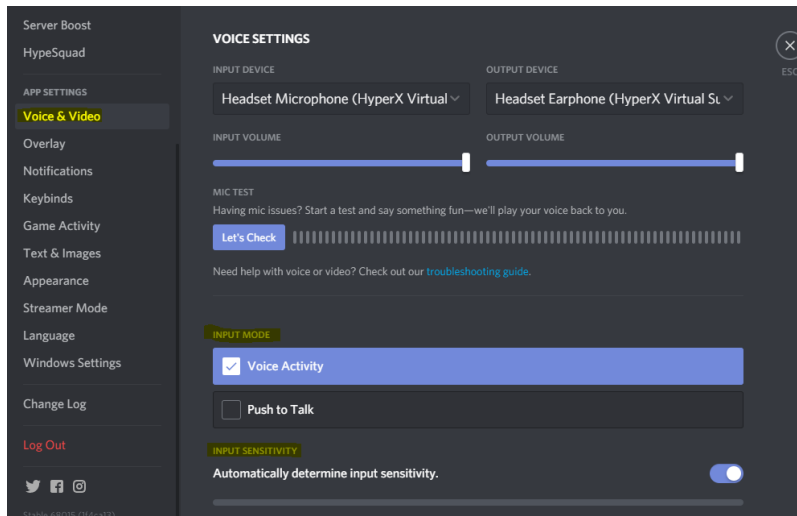
For participants that don’t have a substantial public following, simply disabling DMs from server members and allowing anyone to add you as a friend will likely be sufficient. The above recommended settings are to ensure minimization of undesired contact. Less restrictive settings will make it easier to add others, so it might be worth starting with those and making them more strict as needed. This setup usually works well for me, with only a few intermittent random friend requests.

The options below this area contains additional settings related to privacy between the user and Discord. Since these are purely personal decisions and have no direct impact on the sprint, it will not be covered in this guide.

6.2 Audio settings

The default settings can work reasonably well prior to joining, but I have a few specific recommendations based on my own experience with Discord.

The audio settings can be accessed by clicking the lower left gearbox and going to “App Settings > Voice & Video”.



At the top of the page, it is worth briefly verifying that the input/output devices match the ones that are desired to use, such as selecting your headset instead of built-in audio. Also, be sure to click “Let’s Check” to verify that discord is able to pick up on your voice.

As for the recommended input settings, it depends upon local audio setup.

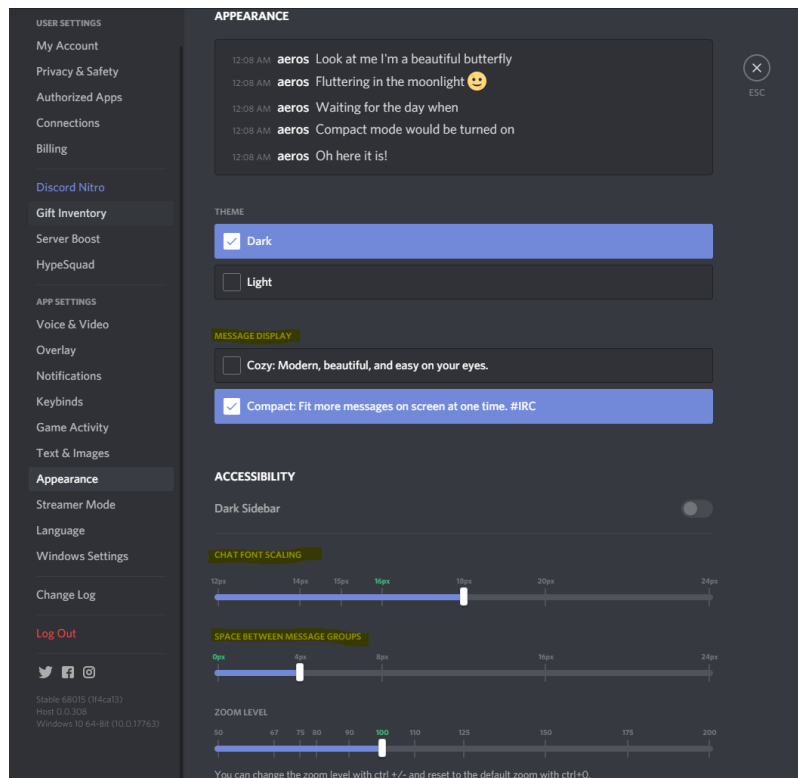
If relying on a built-in area mic such as for a phone or laptop, I would strongly recommend usage of “Push to Talk” (otherwise, it will likely pick up on any background noise). Otherwise, if using a headset (with some noise cancellation) or dedicated microphone (with a pop filter), “Voice Activity” with “Automatically determine input sensitivity” generally works well.

The “Video settings” section below allows you to select an external recording device (such as for face cam), and below that are more advanced audio settings. I generally leave these on the defaults, but experimenting with them may be worthwhile if you are experiencing issues with your voice not being detected (such as disabling “Noise Suppression”, “Noise Reduction”, and/or “Advanced Voice Activity”). For further troubleshooting, I’d recommend [Discord’s guide](#).

6.3 UI settings

This will be the most opinionated part of the guide, and if you are satisfied with the default settings, this part can be safely skipped. However, I personally like to use a compact mode that minimizes extraneous elements (such as profile pictures) to fit more messages on the screen, while still having a font size and spacing large enough to easily read. This is more similar to an IRC style, for those familiar with it.

The UI settings can be accessed by clicking on the user settings gearbox in the lower left and going to “App Settings > Appearance”:



My recommended settings are:

“Message Display”: “Compact”

“Chat font scaling”: 18px

“Space between message groups”: 4px

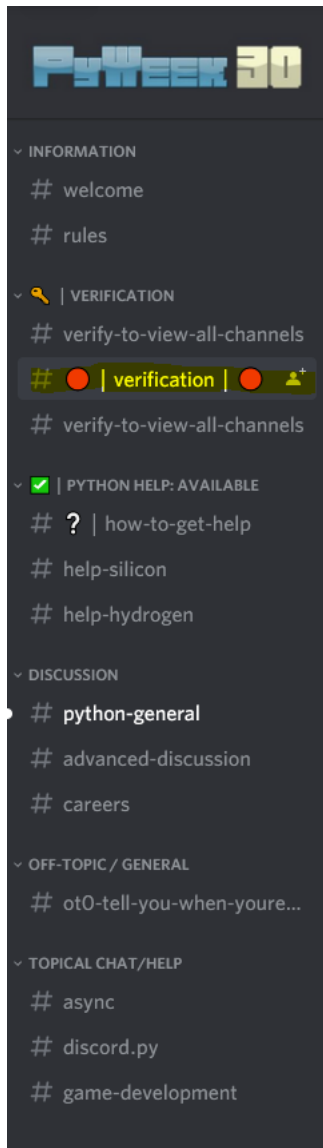
Of course, this can easily be tweaked as needed to increase font size or spacing based on eyesight, screen resolution, and personal preferences. On smaller resolutions, adjusting the zoom size will make it easier to see more channels at once.

6.4 Joining Python Discord

Python Discord (the public server where the sprint is being hosted) can be joined through the following perma-link: <https://discord.gg/Q87A9Y9>.

If you haven’t already, I’d recommend creating a discord account prior to joining, and double-checking that your personal privacy settings are as intended.

When first joining the server, you will see a limited preview until verifying (this is to prevent/limit bot accounts from joining). In order to do so, click on the channel titled “verification”:



Next, simply type “!accept” within the chat and press enter after reading over the server rules (they are very brief and mostly common sense). Welcome to Python Discord!

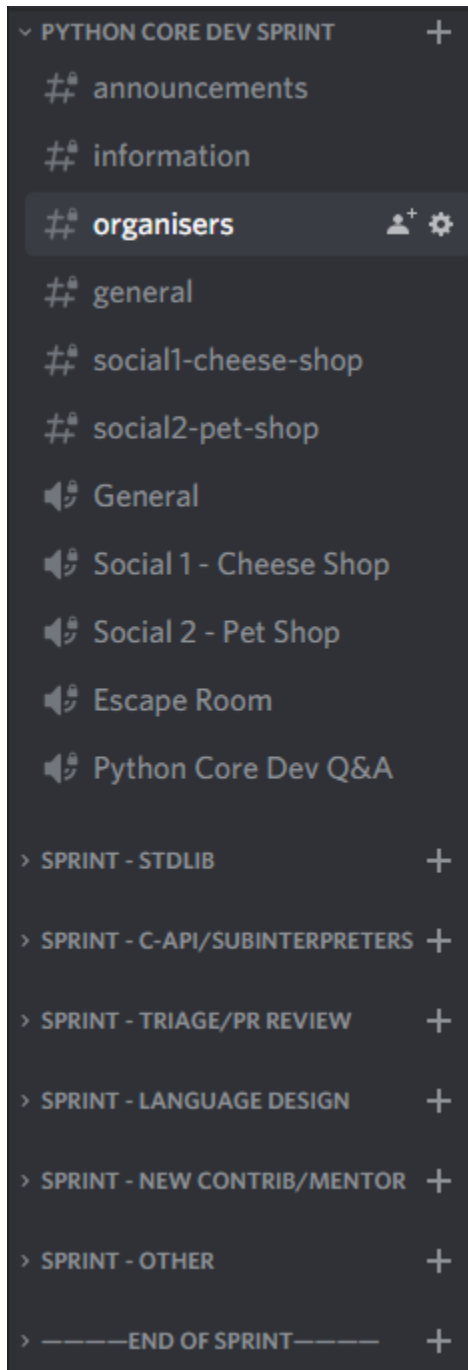
At this point, if it is before the start of the sprint, you will likely see the standard layout of channels within the server that are viewable by all verified users. Feel free to explore around if you’re curious, notably “python-general”, “advanced-discussion”, and the topical channels might be of interest.

Prior to the sprint, all participants will be granted a sprint participant role on the server that will allow them to see the channels we will be using throughout the sprint. In order to do this, an announcement will be sent out to collect IDs ahead of time, and then a script will be used to automatically grant the role to everyone that provided their ID.

However, for this to work, participants must already be on the server (in order to give them the role). If the sprint has already started or you are joining the server late, reach out directly to an organizer of the event (such as myself, at aeros167@gmail.com) and we can grant you the role manually after joining.

6.5 Channel structure & navigation

After being granted the sprint participant role, you will see something like the following channel layout at the top of the server:



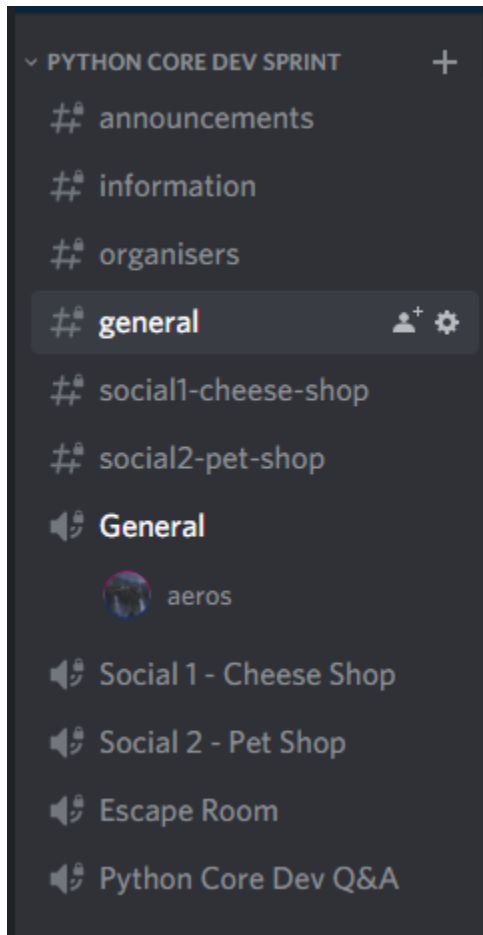
These categories are where the sprint will be primarily taking place, and will only be visible to participants and the admin team at Python Discord (other than the Q&A voice channel, which will be opened to users of the server to join muted on Oct. 20th when it takes place).

Discord uses two primary types of channels: text channels and voice channels. Text channels are indicated by a pound/hash (#) symbol to the left of the name, and voice/video channels are indicated by the audio symbol (). Any

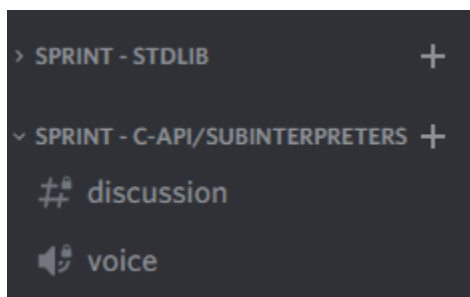
channel can be joined simply by left-clicking on it.

You can simultaneously view a text channel while being present in a voice channel. This is done by simply left-clicking the text channel after joining the voice channel. Typically, the corresponding intended text channel will have a similar name; e.g. “general” for the text channel and “General” for the voice channel, or in a nearby pair like “discussion” and “voice” within a category. This way, those without a mic can listen in and respond via text, or other content like links, images, or code blocks (standard markdown format) can be shared in the text channel.

For example, this is what it might look like while being in the “General” voice channel while simultaneously viewing the “general” text channel (with the contents of the channel appearing in the center of the screen).



The next component to be aware of are categories, which act as a visual means to separate groupings of channels. They can be expanded or collapsed either by left-clicking on individual categories or on platforms where available, **ctrl+shift+a** to toggle collapse/expand all channels.



As for the overall channel layout, the sprint encompasses all categories starting from “Python Core Dev Sprint” all

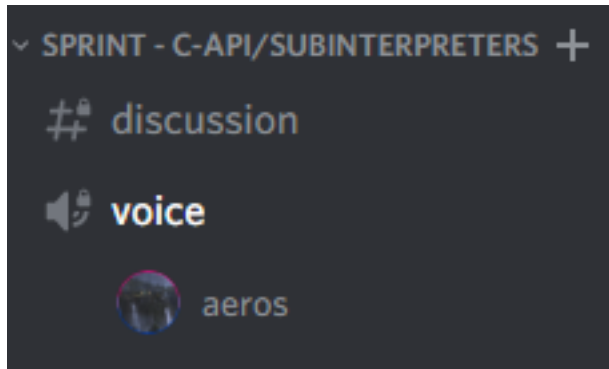
the way down to the separator “END OF SPRINT”. Below that contains the typical channels viewable by all users on Python Discord.

The first category, “Python Core Dev Sprint”, contains the following:



- **announcements** - used for communicating event and sprint project start times as well as general reminders that are relevant to all participants.
- **information** - will contain a link to this guide and contact info of organizers for help.
- **general/General** - used as a general-purpose chat room for all participants and location to ask questions that are not specific to a sprint project.
- **social<n>/Social <N>** - used as channels for general social activities or unstructured chatting. This is a great location to take a break from working on a project and leisurely talk to other participants.
- **Escape Room** - specifically used for participation in the virtual escape room social activity.
- **Python Core Dev Q&A** - used for hosting the community -> core dev Q&A session taking place on Oct. 20th. Note that this channel will be the only publicly viewable and joinable channel by users of the server (muted), so it's not suited for private discussions.

The remaining categories are for individual sprint projects, with each containing at least one pair of voice/text channels to use. For example:

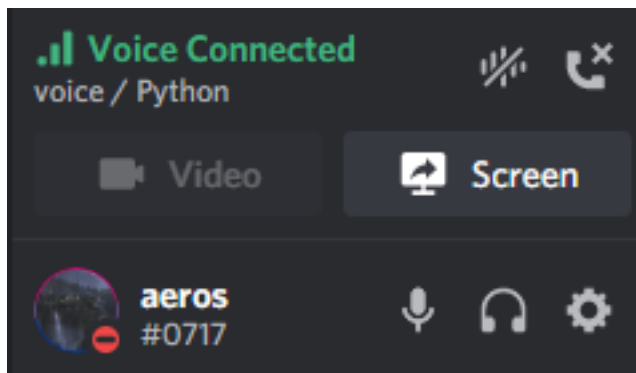


These channels should be used whenever working collaboratively on a specific project, and the category or channel name should reflect the project being worked on. This will make it easy for other sprint participants to easily see what others are working on and change between projects as desired.

The last “Sprint - Other” channel will be used as an overflow category, in case all other sprint project categories are currently occupied.

While working on a specific project, I’d recommend collapsing the other sprint categories. Icons of participants in other voice channels will still be visible, and can be clicked on to see who it belongs to. This helps to minimize overall visual clutter, and easily locate relevant channels.

The last important general channel usage topic are voice channel specific features, such as screen sharing, enabling video, muting, and deafening. After joining a voice channel, the following menu should be visible in the lower left of of your application:



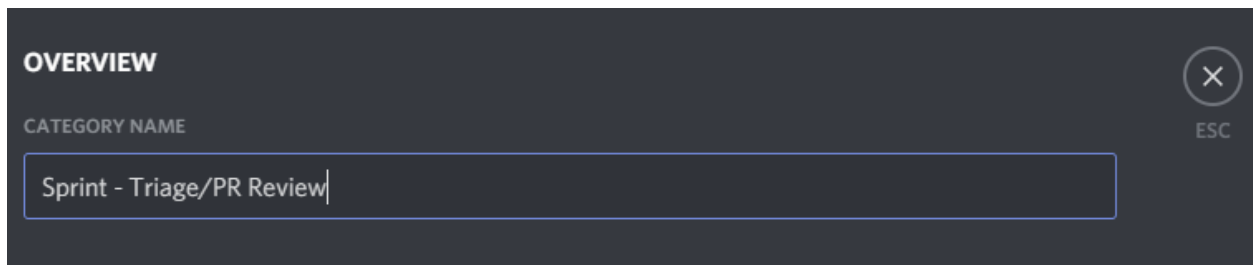
Clicking on “Screen” will open a menu that allows you to screen share a specific application or monitor, which will be visible to other participants in the channel. “Video” can be clicked on for showing a webcam (or other video device). The phone with the x can be used to leave the channel. The mute/deafen ones below it are fairly self-explanatory, with the mic being the universal mute icon and the headphones being deafen.

6.6 Renaming categories & creating new channels

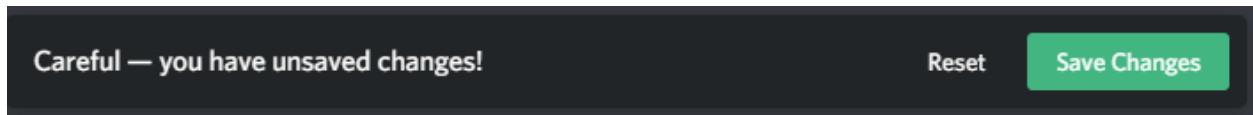
In order to minimize the total amount of visual clutter while still providing enough channels, we created 5 separate categories that can be reused for different projects. The current naming is based on the sprint projects with the most interested participants, but we've provided permissions for participants to easily change the name of each category. This should be done only if there is nobody else present within the voice channels (within the category) and a category for your sprint project is not listed.

Also, **“Sprint - Stdlib” and “Sprint - Other” should NOT be renamed.** “Stdlib” will contain several channel pairs for each module, and “Other” is used as overflow.

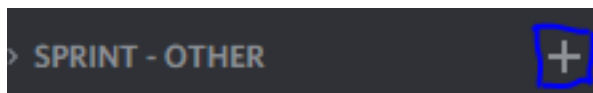
In order to rename a category, simply right click on it and select “Edit Category”. In the center of the screen, the following will appear:

A dark-themed dialog box titled 'OVERVIEW' with a close button (X) in the top right corner. Below the title is a label 'CATEGORY NAME' and a text input field containing 'Sprint - Triage/PR Review'. In the bottom right corner, there is a small 'ESC' label.

Then, after the “Sprint - “, write the name of your project. For example, if you wanted to work on the PEG parser, it could be renamed to “Sprint - PEG Parser”. Then, press “Save Changes” at the bottom to update the name:

A dark-themed bar at the bottom of the dialog. On the left, it says 'Careful — you have unsaved changes!'. In the center is a 'Reset' button. On the right is a green 'Save Changes' button.

If all sprint project categories have participants in their voice channel(s), instead of renaming a category, a new text/voice channel pair can be added within “Sprint - Other”. This can be done by clicking on the “+” icon to the right of the channel name:



Then, the following menu will appear:

CREATE TEXT CHANNEL

in Sprint - Other

CHANNEL TYPE

☒ # Text Channel

☐ 🔊 Voice Channel

☐ 📢 Announcement Channel

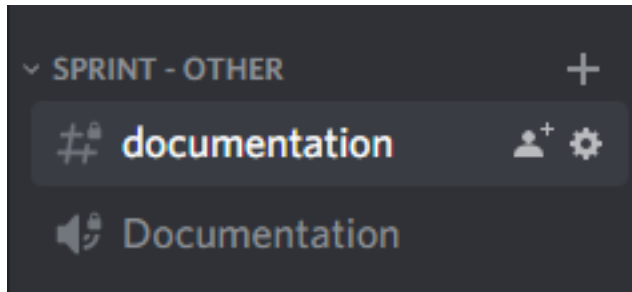
CHANNEL NAME

☒ Private Channel

By making a channel private, only selected roles will have access to read or connect to this channel

Cancel>Create Channel

Within this menu, you can add new text or voice channels. For example, for working on Documentation, you might add a text channel titled “documentation” with a corresponding voice channel titled “Documentation”. This is what it would look like:



Note that the permissions will be inherited based on the parent category, so any newly added channels will have the same restrictions to only being viewable by sprint participants (and server admins).

There's no hard limitation as to the number of channels that can be created, but to keep the total count to a minimum to make things easier to keep track of, we may periodically delete extra channels at the end of the sprint days. So, make sure to record any important information such as meeting notes within the google doc assigned to your project (or create a separate one if there isn't). The doc for each project can be accessed via [Sprint Projects](#).

PYTHON COMMUNITY Q&A SESSION

We have collected questions from the community via [slido](#). We will then assign a question to a core dev before the Q&A start, so the core dev can come prepared with answer.

The community Q&A session will be recorded and published to The PSF's YouTube channel.

Schedule: Tuesday, Oct 20, 2020, 3 PM Central time.

Python community had the opportunity to ask a question to core devs via [slido](#). This was open between **September 24 - September 30**.

Core developers, please go to this [Google Sheets](#) and assign a question to yourself. We suggest prioritize answering the ones with most upvotes, however feel free to pick a question that you're comfortable with answering.

See also:

[Participant List](#) and [Sprint Projects](#) for the list of all sprint participants and their sprint projects

7.1 Core devs participating in the Community Q&A

- Joannah Nanjekye
- Barry Warsaw
- Pablo Galindo Salgado
- Mariatta
- Łukasz Langa
- Senthil Kumaran
- Carol Willing
- Andrew Kuchling
- Neil Schemenauer
- Petr Viktorin, if there's relevant question
- Zach Ware
- Brett Cannon
- Eric V. Smith
- Stefan Behnel
- Ned Deily
- Lysandros Nikolaou

- Emily Morehouse
- Jason R. Coombs
- Kyle
- Terry Jan Reedy
- Guido van Rossum
- Thomas Wouters
- Yury Selivanov, can answer questions re asyncio
- Steve Dower, if there's relevant question
- Christian Heimes
- Eric Snow
- Gregory P. Smith
- Brandt Bucher
- Mark Shannon

SOCIAL ACTIVITIES

Contents

- *Social Activities*
 - *Escape Room*
 - * *Your escape adventure starts here!*
 - *Online Board Game*
 - *Unstructured social time*

8.1 Escape Room

We will be playing the game [Escape From Cell 126](#). ([Online review](#) of the game). We have purchased 6 access codes, so up to 6 teams can play on this game.

If you'd like to join a game, please sign up by filling in this [Google Sheets](#). (You should have access to this if you've signed up for the sprint).

If there are no more spots available and you'd like to play, please let us know!!

Each group is responsible to choose a time to play the game. Allocate at least 1 hour to play it.

Please read the following email ahead of time.

8.1.1 Your escape adventure starts here!

Thank you for purchasing ESC From Cell 126. We can't wait for you to get escaping! Make sure you read this email carefully. It includes **important information** about the game and **your unique game link**.

The important bits...

Your team

You can have as many players in your team as you like but we think the game works best with 2-6 people. If you are playing with a remote team, you will need to set up a video call so you can talk to each other. We recommend using an app with the ability to screenshare such as Zoom, Google Meet, or Skype.

VERY IMPORTANT!!

If you are using the same device to play the game and video call, **do not block the right hand side of your screen** - this is where your clues are stored and you might miss them!!

Remote teams: to screenshare... or not to screenshare

You have two options when it comes to playing with a remote team:

Option 1) You, as the organiser, go to the game link on your laptop or PC and screen share the game with your team. Get your team to go to the game link on their phones so they can read clues independently.

Option 2) Your whole team goes to the game link on their laptop or PCs. This allows everyone to see the full game and interact with clues and puzzles. Just like in a real escape room, when one person solves a puzzle, it is solved for everyone. Similarly, if one person asks for a clue, the clue is shown to everyone.

We like option one for teamwork but option two means everyone gets to play around with the puzzles. The choice is yours!

Your device and browser

A desktop or laptop is ideal but ultimately, you'll need a device which is at least as big as an iPad in order to view the full game. You can view the clues on any device. Check whether your device is compatible by going to this [link](#).

You should use the latest version of Google Chrome or Firefox to play the game on your laptop or PC. Other browsers may work but we cannot guarantee that all features will be supported. If you are using an iPad, we recommend using Safari.

What if I get stuck?

There are three hints available for every puzzle within the game. The hints will progressively give you more information. If you still do not have the correct answer after three hints, you will be able to view the answer.

Timings

You can start the game when you like. There is no time limit but most players finish the game in around 1 hour.

Can I play it again?

Your game link is unique and can only be used once. If you accidentally leave your game mid session, you can come back to it using the same link - all your progress will be saved.

The exciting bits...

Your unique game links:

We will populate the [Google Sheets](#) with the game links prior to the start of the sprint.

Good luck and happy escaping!

ESC The Game Team

8.2 Online Board Game

Details TBD

8.3 Unstructured social time

There will be a dedicated “break” room in Discord. Feel free to jump in there and chat with other attendees.

TECH CHECK SESSION/TECH DAY

On Oct. 18th (Sunday) from 3pm to 12am EST (UTC-4), there will be a tech check session hosted by Kyle Stanley. The purpose of this session will be for participants of the sprint to verify ahead of time that they have access to Python Discord (for the sprint channels), Zoom, working voice/audio, etc.

During the period of 3pm to 12am EST, Kyle Stanley will be specifically available to answer any setup-related questions, either through text or voice/video call. Here are the preferred methods of contact:

Email: aeros167@gmail.com

Discord: aeros#0717 (send friend request first)

For voice/video calls, use either Discord call or Google hangout call (Discord preferred).

For any Discord setup related questions, make sure to view the [Discord Setup Guide](#) prior to reaching out.