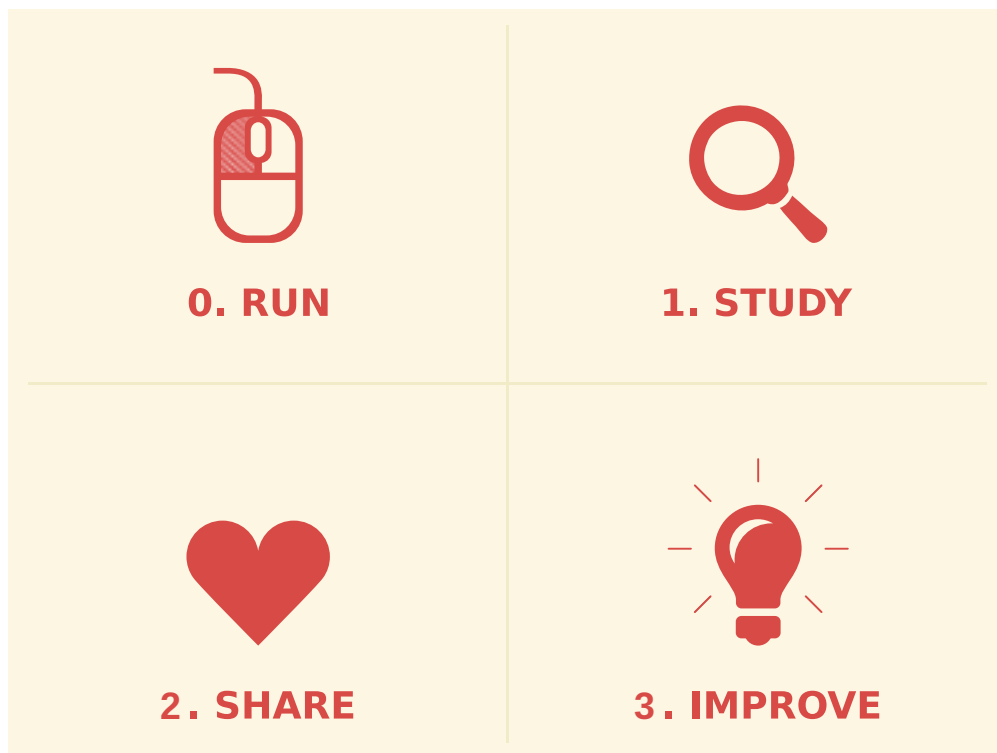


Desenvolvimento Aberto



Apresentação da Disciplina + Workflow de trabalho distribuído

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)

Burocracias

Horários de aula:

- TER 09:45 - 11:45
- QUI 09:45 - 11:45

Atendimento:

- TBD

Avaliação

- Curso baseado em projetos feitos individualmente e/ou com apoio de um grupo.

Objetivos de Aprendizagem

Ao final da disciplina o estudante será capaz de:

- Analisar uma base de códigos desconhecida de médio/grande porte e modificá-la de modo a fazer melhorias e corrigir falhas em um software;
- Interagir com uma equipe remota de desenvolvedores para entregar código que atenda aos padrões de qualidade e estilo de código de um projeto;
- Entender as diferenças licenças de software livre e como elas impactam na distribuição e reutilização de uma base de código.

Objetivos (versão resumida)

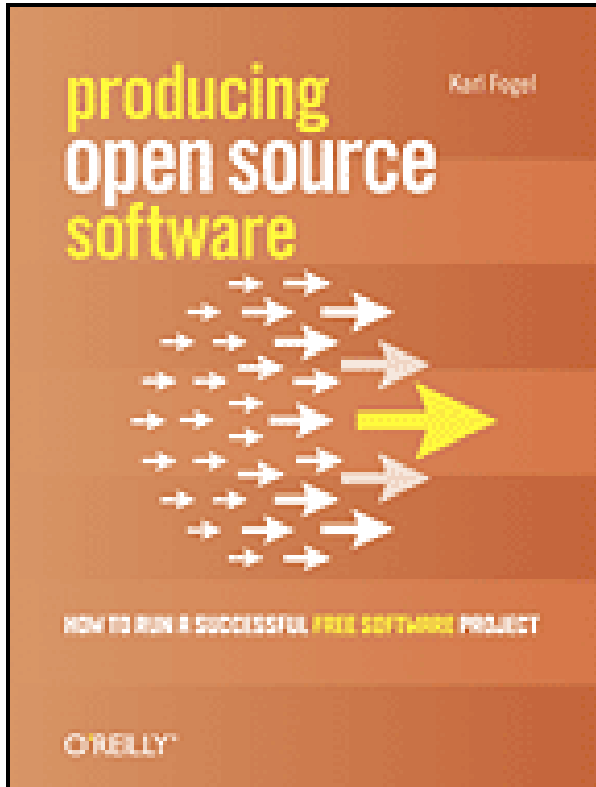
Ao final da disciplina o estudante será capaz de:

- Baixar, entender e **modificar** o código de um projeto
- Conseguir que suas modificações sejam **aceitas pelo projeto original**
- Compreender aspectos ligados a **distribuição** de software.
 - Licenças
 - Bug Tracker, Versionamento, Governança, etc
 - Documentação / Internacionalização
 - Comunidades de usuários

Programa do curso

1. Modelos de desenvolvimento e comercialização de software;
2. Licenças de software e seu impacto na reutilização e distribuição;
3. Ferramentas de apoio ao desenvolvimento colaborativo de software (livre ou proprietário);
4. Documentação de software e de código;
5. Tradução e internacionalização de Software
6. Sistemas de compilação e distribuição de código fonte;
7. Aspectos humanos e comunitários em desenvolvimento de software;
8. Estudo de casos de sucesso.

Livro texto



Disponível online em <https://producingoss.com>

Materiais do curso

Github: <https://github.com/insper/dev-aberto>

Site: <https://insper.github.io/dev-aberto/>



Blackboard será usado para avisos somente. Todo conteúdo estará disponível no github.

Justificativa da disciplina

Nas disciplinas anteriores trabalhamos

- criando um projeto novo.
- que normalmente morre após a disciplina
- e nunca é usado por ninguém

No mercado, normalmente trabalhamos em um projeto existente

- corrigindo problemas
- realizando melhorias
- que é usado por vários usuários

Por que vocês estão aqui?

<https://forms.gle/UzvEtYKJxQBtm9f29>

Resultados alcançados nesta disciplina até agora

Vitória - 2018/2

Spyder

- Aceito: [\[1\]](#)

Pandas

- Aceito: [\[1\]](#)





Paulo - 2018/2

Cataclysm: DDA

- Aceito: [\[1\]](#)

Turma - 2019/2

15 estudantes

19 projetos diferentes, 14 PRs aceitos

Pandas - 6 contribuições aceitas

 [pandas-dev](#) / [pandas](#)

 Sponsor

 Used by

175k

 Watch

1k

 Star

23.4k

 Fork

9.3k

 Code

 Issues 3,145

 Pull requests 150

 Actions

 Projects 3

 Wiki

 Security

 Insights

BUG: fix custom xticks in bar() plotting function #28172

 Merged

TomAugspurger merged 7 commits into [pandas-dev:master](#) from [GiuPassarelli:xticks_bug](#) on Sep 23, 2019

 Conversation 16

 Commits 7

 Checks 12

 Files changed 3

+15 -2 



GiuPassarelli commented on Aug 27, 2019

Contributor

+  ...

- ☐ closes [#14119](#)
- ☐ tests added / passed
- ☐ passes `black pandas`
- ☐ passes `git diff upstream/master -u -- "*.py" | flake8 --diff`
- ☐ whatsnew entry

Reviewers

-  WillAyd 
-  TomAugspurger 

Assignees

No one assigned

Labels

- Bug
- Visualization

  Fixing issue [#14119](#)

 2bcf920

Turma 2020/1

16 estudantes

15 projetos diferentes, 17 PRs aceitos

Pandas, Matplotlib, Bokeh, Pygame

Adds normalize kwarg to pie function #16985

Open with

 Merged timhoffm merged 6 commits into matplotlib:master from raphacosta27:issue-#16905 on May 2, 2020

 Conversation 45  Commits 6  Checks 15  Files changed 4 +60 -5



raphacosta27 commented on Mar 31, 2020 • edited

Contributor






PR Summary

pie() used to draw a partial pie if the sum of the values was < 1 and already normalized the values if the sum was > 1. I added a kwarg normalize into pie function and deprecated the previous behaviour. Now, the function will always normalize the values to a full pie by default. If one wants to draw a partial pie, one should pass normalize=False explicitly.



PR Checklist

- ☒ Has Pytest style unit tests
- ☒ Code is Flake 8 compliant

Reviewers

-  tacaswell ✓
-  QuLogic
-  brunobeltran
-  timhoffm ✓
-  jklymak

Assignees

-  tacaswell
-  timhoffm

Apresentação do curso

Apresentação do curso

Proposta: Curso será *gamificado*

- Cada aluno criará um *avatar* na disciplina
- Atividades para entrega valem XP
- Cada atividade é representada por uma *skill* com um nome engraçadinho.
- Toda entrega de trabalho é via *Pull Request* no repositório da disciplina.
- **Grande liberdade de escolha**

Apresentação do curso (Skills)

- **Tutorial:** atividades preparatórias (guiadas) em sala de aula
- **Código:** contribuições de código para projetos externos
- **Comunidade:** contribuições não técnicas que podem beneficiar usuários e desenvolvedores de um projeto, como documentações e traduções
- **Impacto:** resultados significativos alcançados pela participação do aluno em projetos abertos

[Lista completa de skills](#)

Apresentação do curso (plano de aulas)

- Agosto/Setembro: **Tutorial**
 - 30-60 minutos de expositiva/discussões
 - Atividades focadas em um tema específico
- Outubro/Novembro: Projeto
 - **Autonomia** para definir quais tarefas serão feitas
 - Escolha de projetos será semi-livre

Apresentação do curso (avaliação)

- Todas as entregas são individuais
- Cada objetivo de aprendizagem é medido por várias skills.
 - Algumas são obrigatórias
 - Repetir uma skill pode valer mais ou menos pontos que a primeira vez.
- Nota é baseada na quantidade de *XP* obtida.
 - Liberdade para decidir onde investir tempo/esforço.

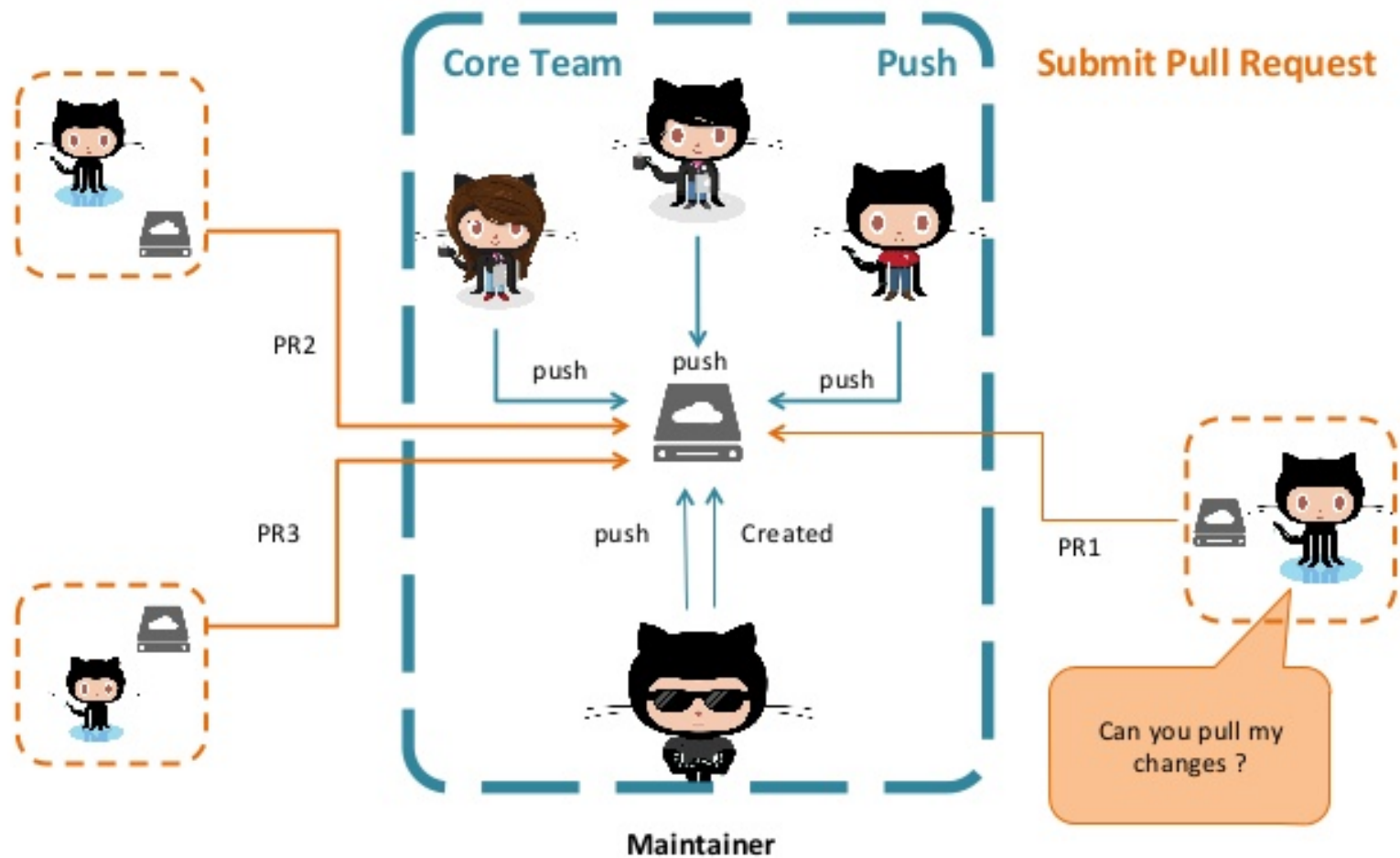
Apresentação do curso (avaliação)

- Nota final é uma combinação de XP e skills obtidas
- Relatório quinzenal enviado por email
- Condições completas na [página de regras e skills](#)

Discussão: workflow de desenvolvimento

Como Git funciona?

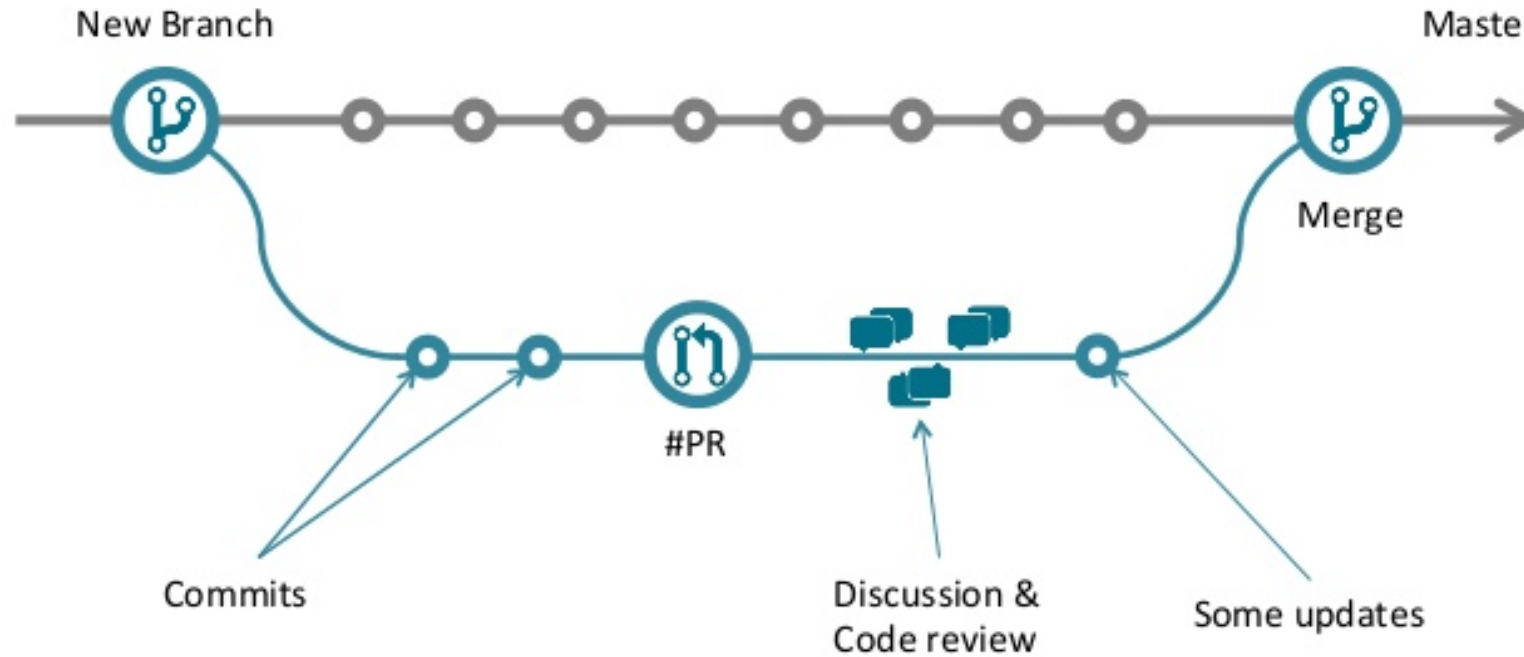
Pull Request Model



Sources : <https://www.iconfinder.com> & <https://octodex.github.com/>

Fonte: <https://www.slideshare.net/abderrahmaneбенbachir/continuous-integration-in-github>

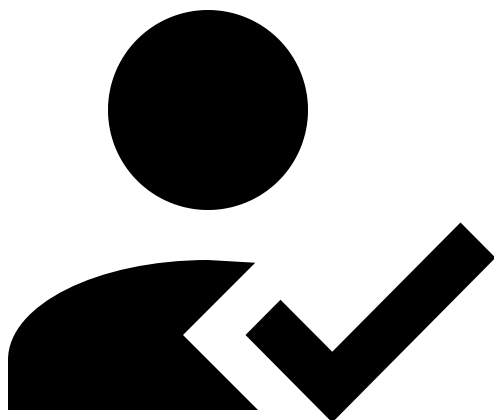
Pull Request Process



Sources : <https://www.iconfinder.com> & <https://octodex.github.com/>

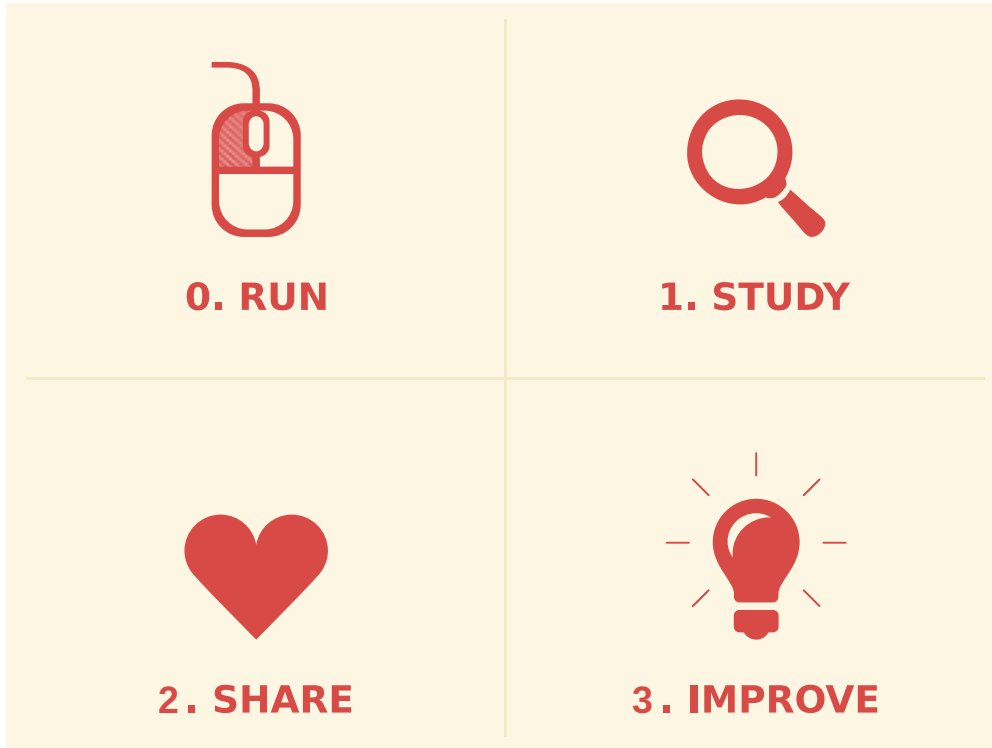
Fonte: <https://www.slideshare.net/abderrahmanebenbachir/continuous-integration-in-github>

Atividade: Primeiros passos



Objetivo: Enviar seu primeiro *Pull Request* para o repositório da disciplina.

Desenvolvimento Aberto



Apresentação da Disciplina + Ciclo de vida de um Bug

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)