7 월 21 일 안내

—

안녕하세요. 발대식 및 컨트리뷰톤 진행 방향을 안내드립니다.

그간 프로젝트 셋업과 간단한 이슈들 보시면서 몸풀기가 되셨으면 좋겠습니다 :)

먼저 발대식은 7 월 25 일 토요일 오후 4 시이고, 장소는 **Open UP** (서울 강남구 테헤란로 431, 저스트코타워 13 층) 컨퍼런스룸 입니다.
날짜는 참석 가능여부 설문을 바탕으로 선정하였습니다.
행사 중 간단한 다과 등이 제공된다고 하니 참고 부탁드립니다.
아직 빌드 및 테스트 실행을 완료하지 못한 분들은 발대식 참가 전 반드시 이전에 드린 안내에 따라 빌드와 테스트 실행을 해 주세요. 개발 환경 준비에 문제가 있다면 꼭 알려주셔야 합니다.

RustPython 컨트리뷰톤은 각자 프로젝트에서 기여하고 싶은 주제를 찾으시면, 필요한만큼(~=요청하시는 만큼) 기술적인 지원을 해드리는 방식으로 진행이 되는 것이 기본입니다.
RustPython 프로젝트에서는 많은 중소형 오픈소스 프로젝트와 마찬가지로 정해진 개발 목표나 특정 태스크가 특정 사람에게 할당되지 않습니다. 프로젝트 운영에 어울리게 각자 원하는 주제를 찾아 주시되, 탐색에 어려움이 있으면 도움을 요청해 주세요.

RustPython 컨트리뷰톤은 gitter 를 중심적으로 사용합니다. 온라인 활동에 어려움을 느끼는 분들을 위해 중간에 오프라인 모임을 몇 번 제공할 예정이지만, 여전히 gitter 가 중심이라는 점을 기억해 주세요. 오프라인 모임은 많아 봤자 몇 번 내로 열릴 예정이고, 온라인 커뮤니케이션에 어려움을 느끼는 분들께 막막한 부분 한두 군데를 뚫는 데는 도움이 되지만 온라인으로 지속적으로 의견을 나누면서 얻을 수 있는 빠른 피드백 주기와 링크 등의 레퍼런스 활용을 따라갈 수 없습니다.
RustPython 프로젝트 역시 github issues 와 gitter 를 주요 커뮤니케이션 창구로 채택하고 있습니다.

현재 RustPython 프로젝트에서 핵심 목표로 추진하는 주요 목표는 다음과 같습니다.
PIP 지원: https://github.com/RustPython/RustPython/issues/1983
CPython 표준 라이브러리 이식하기: https://github.com/RustPython/RustPython/issues/8
CPython 테스트 커버리지 높이기: https://github.com/RustPython/RustPython/issues/1671

이 가운데 표준라이브러리/테스트커버리지는 이전에 공유드린 'unittest 로 기여하기' 문서를 바탕으로 쉽게 기여할 수 있습니다.

돌려보고 싶은 내 코드가 있다면 내 코드를 직접 돌려보면서 찾은 문제를 고치셔도 좋아요. 이 방법은 오랫동안 권장되는 버그 탐색 방법이었는데, 최근에

기능 커버리지가 약간 올라가면서 쉬운 코드에서는 버그를 찾기가 어려워져 우선순위를 약간 낮추었습니다. RustPython 의 목표는 실제 파이썬 코드를 돌리는 것이므로 이 방법으로 찾는 버그를 고치는 것은 언제나 좋은 방법입니다.

그 외 등록된 이슈 가운데 난이도가 그리 높지 않은 이슈의 목록을 함께 공유해 드립니다. 등록된 이슈만 검토한 것이고, 테스트 파일을 살펴보면 유사한 이슈를 훨씬 더 많이 발견할 수 있으니, 테스트 파일을 살펴보시면 좋습니다.
빠진 __ne__ 구현하기: https://github.com/RustPython/RustPython/issues/1442
-기본적으로 표준라이브러리 지원과 비슷한 난이도의 단순 기능추가입니다
nonlocal 오류: https://github.com/RustPython/RustPython/issues/2008
-기능 추가보다는 어렵지만 문법 이슈 가운데는 쉬운 편에 들어갑니다. nonlocal 사용시 오류를 즉시 내는 대신 스코프 끝날때 검사하도록 변경하면 됩니다.
pystruct_sequence 의 repr 개선: https://github.com/RustPython/RustPython/issues/1448
-proc macro 로 생성하는 코드를 변경해야합니다. 이미 rust 코딩 경험이 있으신 분이 매크로를 해보고 싶다면 추천드립니다.
소수점 자릿수 지정 반올림 알고리즘
수정: https://github.com/RustPython/RustPython/issues/1554
-파이썬과 관련된 내용은 적고 대부분 부동소수점 알고리즘 문제입니다. 부동소수점이나 수 연산을 다루는데 익숙한 분께 권해드리는 알고리즘 난이도 상
부동소수점 문자열 포맷 형식
추가: https://github.com/RustPython/RustPython/issues/1656
-정확한 포맷을 하는건 어려울 수 있지만 새 포맷을 추가하는 건 어렵지 않게 할 수 있습니다. 문자열은 자유롭게 다룰 수 있는 편이 좋습니다.
lz4 구현체 교체하기: https://github.com/RustPython/RustPython/issues/1909
-rust 에서 외부 crate 를 쓰는 정도 수준의 rust 미숙련자만 아니면 해볼만한 문제
os.exec* 함수 추가: https://github.com/RustPython/RustPython/issues/1992
-liinux,macos 사용자만 가능합니다. nix crate 에서 함수콜 이어붙이는걸로 가능한 표준라이브러리 지원 이슈
입력 숫자에 따른 pow 오동작
수정: https://github.com/RustPython/RustPython/issues/1986
-파이썬의 pow 함수는 사실 여러 지수연산을 한 데 합쳐 놓았기 때문에 rust 에서는 케이스별로 나누어 대응해야 합니다. 누락된 브랜치를 만들면 해결

스스로 이슈를 찾기 어려우신 분은 본인의 관심사를 알려주시면서 이슈를 찾아달라고 요청해 주세요. 스스로 찾는 이슈만큼 만족스럽지는 않을 수 있지만 해결할 수 있는 수준의 이슈를 찾아드리도록 노력하겠습니다.

Rust 관련 질문을 할 수 있는 한국어 러스트 사용자 그룹을 함께 소개해 드립니다: https://rust-kr.org
discord 채팅방이 주력입니다.

마지막으로 한 번 더 강조드리면,
궁금한 점은 사소한 것이라도 무엇이든 gitter 메시지로 질문 주세요. “시간이 날 때 천천히 해결해야지” 보다는 일단 물어보는게 낫습니다.