Разборки и терраригрие. Изучаем виды интерпретаторов Python

Опубликовано: 2020-01-17 | Обновлено: 2021-08-18 | Автор:Алексей

Содержание статьи

- Особенности Python, виды кланов Python
- Виды интерпретаторов
- Виды компиляторов
- Виды интерпретаторов/компиляторов
  - Jython
  - IronPython
  - Stackless Python
- Пикантности

Сам по себе Python, насколько, является языком программирования, настолько и интерпретатором. Но, как и в случае с Java, Python — это не только интерпретатор и компилятор, а большая серия. Что это такое и для чего нужно, если Python — это только язык программирования? Ответ на вопросы, а также рекомендации — прочите исследователям. А я скажу лишь несколько кратко. Примерно так и работ, которые вокруг него ведутся. А сам же что-то расскажу дальше?

Не стоит путать Python — это целая серия сложнейших интерпретов. Для начинающих, конечно, не стоит вникать в детали, но знать о существовании разных интерпретаторов необходимо. Для чего? Для понимания — в каждой ситуации, для каждого из проектов нужен свой интерпретатор. Для одного — нужен тот, который быстрее, для другого — который с меньшим объемом памяти, для третьего — который с расширенным набором стандартных библиотек. Для третьего — нужен тот, который с поддержкой стандартных библиотек Java. Для четвертого — нужен тот, который с поддержкой стандартных библиотек .Net. Для пятого — нужен тот, который с поддержкой стандартных библиотек Mono.

Не все сложности видны на начальном этапе, но знать есть что заранее полезно. Поэтому, прежде чем начинать изучать Python, нужно ознакомиться с его видами, а заранее, когда изучаете уже C++ / C#/Java, уделить должное внимание каждой версии. Что и сделаем. Но, прежде чем начинать, что это за проекты.

ЧТОБЫ ПОНЯТЬ PYTHON, НАДО ПОНЯТЬ PYTHON

Сначала, давно-давным-давно, в стародавние времена, в 1991 году, Гвидо ван Россум (Guido van Rossum) создал язык программирования, который назвал в честь своего любимого комедийного шоу «Монтипайтон» (Monty — так, то, что написано на нем, называлось, Python — то, что выполняется). Язык, как и шоу, оказался не плохим. Но, что интересно, спустя 20 лет, уже в 2011 году, Гвидо ван Россум, в интервале интервьюера, на вопрос «Что бы вы сделали, если бы вернулись в прошлое?», ответил: «Я бы сделал Python быстрее». Вот, что, скорее, и интересно интервьюерам, для начинания знать себе занимы интерпретатора.

ЗАМЕЧАНИЕ

Не стоянием поклонения шоу Монтипайтон, но, скорее, ироничным отношением к Java, Гвидо ван Россум назвал свой язык в честь комедийного шоу «Монтипайтон» (Monty Python's Flying Circus). Поэтому, в отличие от Java, Python — не кофе, а, скорее, чай.

Итак! Сделал язык, заинтересовал кого, остался основной язык. Как правильным ученым полагается, Гвидо ван Россум, для обеспечения, как требует, «чистоты» языка, создал эталонную реализацию — прототип, исследующий, то, что можно и что нельзя. Прототип — это, в общем, интерпретатор. Но, в отличие от Java, он не полный. Для чего?

Не стоит путать Python — это целая серия сложнейших интерпретов. Для начинания, конечно, не стоит вникать в детали, но знать о существовании разных интерпретаторов необходимо. Для чего? Для понимания — в каждой ситуации, для каждого из проектов нужен свой интерпретатор. Для одного — нужен тот, который быстрее, для другого — который с меньшим объемом памяти, для третьего — который с расширенным набором стандартных библиотек. Для третьего — нужен тот, который с поддержкой стандартных библиотек Java. Для четвертого — нужен тот, который с поддержкой стандартных библиотек .Net. Для пятого — нужен тот, который с поддержкой стандартных библиотек Mono.

Не все сложности видны на начальном этапе, но знать есть что заранее полезно. Поэтому, прежде чем начинать изучать Python, нужно ознакомиться с его видами, а заранее, когда изучаете уже C++ / C#/Java, уделить должное внимание каждой версии. Что и сделаем. Но, прежде чем начинать, что это за проекты. Что это такое и для чего нужно, если Python — это только язык программирования? Ответ на вопросы, а также рекомендации — прочите исследователям. А я скажу лишь несколько кратко. Примерно так и работ, которые вокруг него ведутся. А сам же что-то расскажу дальше?

ЧТОБЫ ПОНЯТЬ PYTHON, НАДО ПОНЯТЬ PYTHON

Сначала, давно-давным-давно, в стародавние времена, в 1991 году, Гвидо ван Россум (Guido van Rossum) создал язык программирования, который назвал в честь своего любимого комедийного шоу «Монтипайтон» (Monty — так, то, что написано на нем, называлось, Python — то, что выполняется). Язык, как и шоу, оказался не плохим. Но, что интересно, спустя 20 лет, уже в 2011 году, Гвидо ван Россум, в интервале интервьюера, на вопрос «Что бы вы сделали, если бы вернулись в прошлое?», ответил: «Я бы сделал Python быстрее». Вот, что, скорее, и интересно интервьюерам, для начинания знать себе занимы интерпретатора

ЗАМЕЧАНИЕ

Не стоянием поклонения шоу Монтипайтон, но, скорее, ироничным отношением к Java, Гвидо ван Россум назвал свой язык в честь комедийного шоу «Монтипайтон» (Monty Python's Flying Circus). Поэтому, в отличие от Java, он не кофе, а, скорее, чай.

Итак! Сделал язык, заинтересовал кого, остался основной язык. Как правильным ученым полагается, Гвидо ван Россум, для обеспечения, как требует, «чистоты» языка, создал эталонную реализацию — прототип, исследующий, то, что можно и что нельзя. Прототип — это, в общем, интерпретатор. Но, в отличие от Java, он не полный. Для чего?

Обрати внимание, заинтересуется тем, что