

В какой момент мы читаем кавычки?

Вызов какой-то команды это тоже текст. Но в кавычках оно воспринимается больше как единое слово.

Даже если в двойной кавычке команда.

Получив строку полную сначала всегда trim.

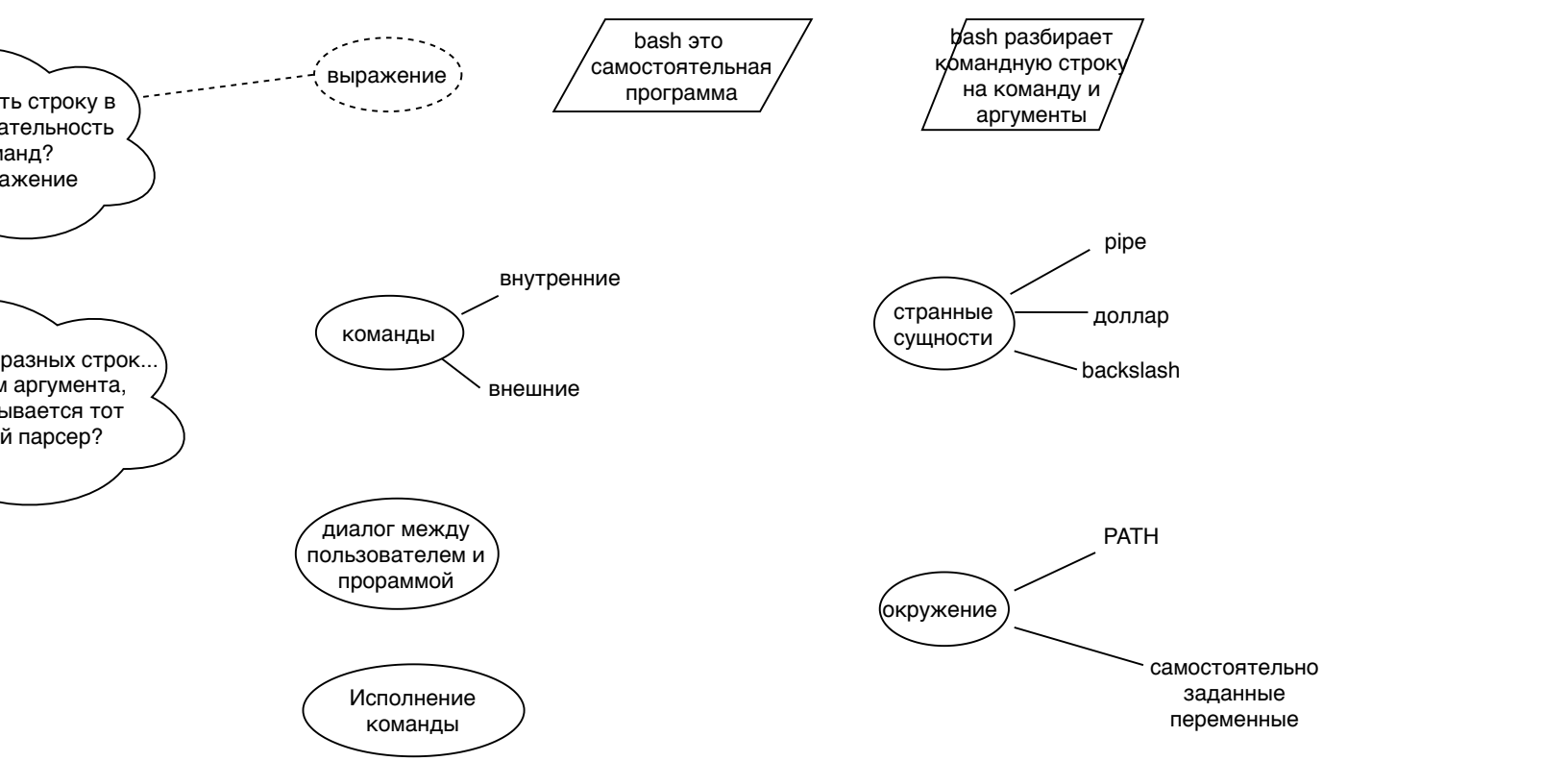
Нет смысле делать переменную глобальной.

Тогда потом разбить на токены. Ридером, почему нет.

Если известно, что какой-то прежуток без кавычек, то там могут быть | например

преврати  
последов  
ком  
в выр

парсеры для  
идет прием  
затем выз  
или ино



словарь

	a: apple	
--	----------	--

Воспринимает всё как слова, чтобы вывести их на экран.

**Вывести свои аргументы на экран**

! ему плевать сколько пробелов между разделителями  
Значит, там мб много аргументов

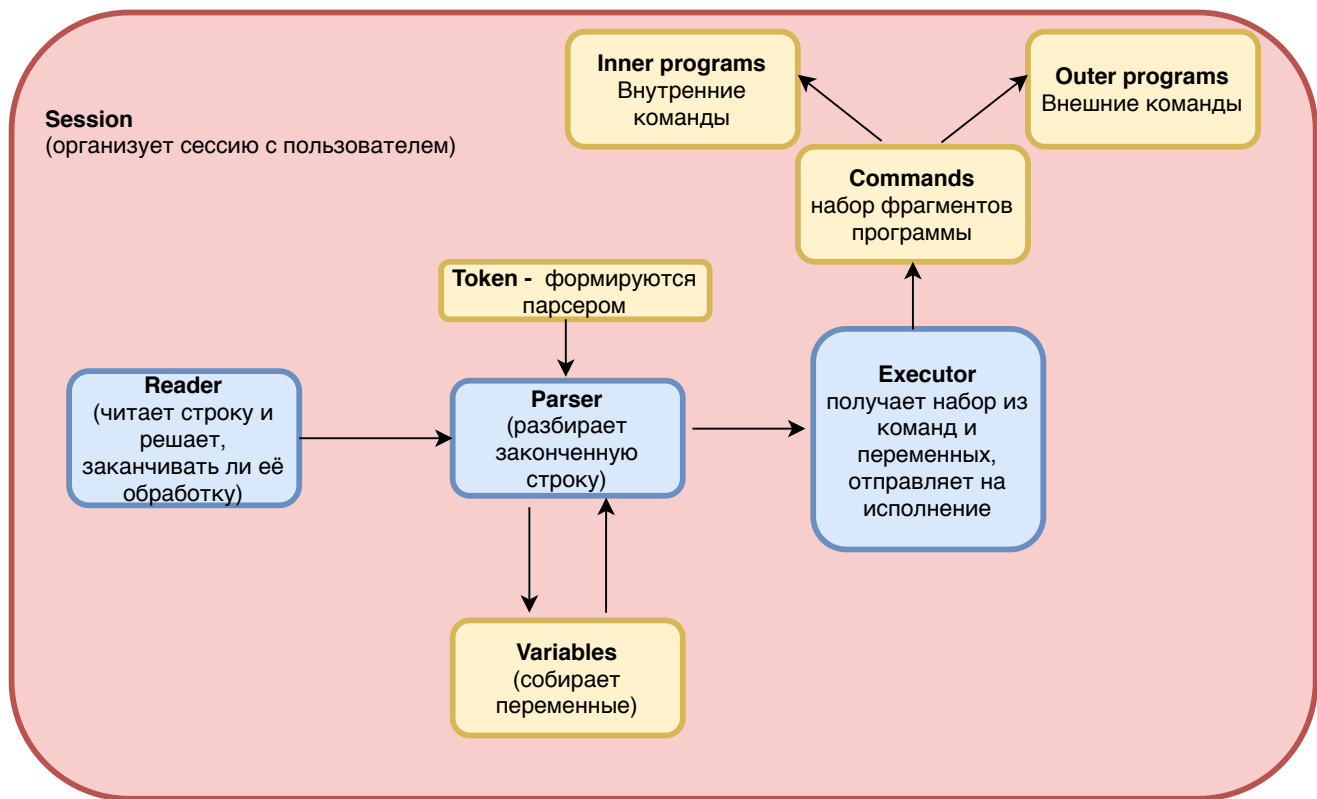
**вывести содержимое файла**

вывести информацию о  
файле

**распечатать текущую  
директорию**

**выйти из интерпретатора**

если это сделать, терминал закроется.





Во что он превращает строку.

После команд echo они раскрываются и печатаются. Причем ' говорит читать буквально и не обращать внимания что там. Проверим.

Токен вытаскивается со всем вариантом строк.

#### Разделители:

пробел, табуляция, перевод строки. Но не внутри кавычки.

Чтобы разделитель попал внутрь слова, окружить его одинарными или двойными кавычками. Тогда она становится одним параметром.

[http://heap.altlinux.org/alt-docs/modules/shell\\_intro/index.html](http://heap.altlinux.org/alt-docs/modules/shell_intro/index.html)

Java - выделенные особенности.

Если раскидать по разделителям, то то что в кавычках пойдет как разная строка

#### Кавычки.

Они нужны чтобы воспринимать то что в них написано как единое слово. Что может быть внутри кавычек?

Если в аргументе echo есть кавычки, то он раскроет доллары в них и заменит на имя переменной.

```
some text
mkarpenko@ubuntu:~$ echo "some text"      "another text"
some text another text
mkarpenko@ubuntu:~$ "echo" "some text"      "another text"
some text another text
mkarpenko@ubuntu:~$ "echo" "some text""another text"
some textanother text
```

Зачем нужны кавычки и как они мешают или помогают читать аргументы?

Теперь токен - это строка. Она может содержать пробел, если находится внутри кавычки. Если две строки написаны абсолютно рядом без пробела, это конечно одна строка.

Внутри двойных примеры:

```
mkarpenko@ubuntu:~$ "echo" "some text"
some text
mkarpenko@ubuntu:~$ "echo" "some text"
some text
mkarpenko@ubuntu:~$ "echo" "some text"
some text
mkarpenko@ubuntu:~$ "echo" "some text"
some text
mkarpenko@ubuntu:~$ "echo" "some text"
some text
mkarpenko@ubuntu:~$ "echo" "some text"
some text
```

Что внутри чего понять сложно, но и искать закрывающую скобку д

--	--

Аргументы и команда  
Имя команды всегда идет первым. Затем может встретиться присваивание.

Но главное - разобрать строку на отдельные токены и выдать их списком.  
Варианты: парсить линейно

проблемы - отделить кавычки друг от друга.  
Но pipe тоже может стоять рядом.

Парсить руками на токены. Можно ли new line в джаве?  
Продолжим читать, пока не закроются все скобки.

```
e text"an\"jhjh"  
e text"an'jhjh"  
e text"an\"jhjh"  
e text"an\\jhjh"  
e text"an'jhjh"
```

верное, придется идти сначала строки  
для данной.