

ЛЕКЦИЯ 17

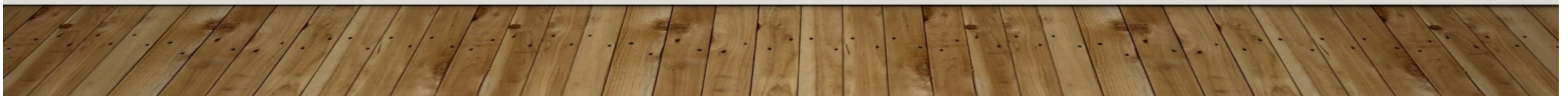
ИНТЕРАКТИВНАЯ КОНСОЛЬ RUBY

ПЛАН ЗАНЯТИЯ

1. Знакомство с IRB
2. Принципы работы IRB

ИНТЕРАКТИВНАЯ КОНСОЛЬ RUBY

- **IRB** — interactive ruby shell или по-простому «**интерактивная руби консоль**». Очень полезный инструмент любого рубиста. Смысл его в том, что вы можете писать любой руби код и видеть результаты его выполнения сразу же, в том же окне.
- **IRB** экономит время, когда нужно написать очень короткую и простую программу, когда нужно понять как будет работать тот или иной код, или быстро вычислить какое-то выражение (как на калькуляторе).



ПРИНЦИП РАБОТЫ

- Для запуска интерактивной рубли консоли достаточно в вашей командной строке набрать **irb**. Результат будет выглядеть примерно так:

```
user#ubuntu:~ aristofun$ irb  
3.0.1 :001 >
```

- Вы попали в интерактивный режим. Вместо **3.0.1** может быть написана другая версия рубли или что-то вроде **irb(main)**.
- Это не важно, важно, что после символа **>** консоль приглашает вас к вводу кода.

ПРИНЦИП РАБОТЫ

- Если вы введете какой-то код, и нажмете **Enter**, то...

```
3.0.1 :001 > puts "Hello world"  
Hello world  
=> nil  
3.0.1 :002 >
```

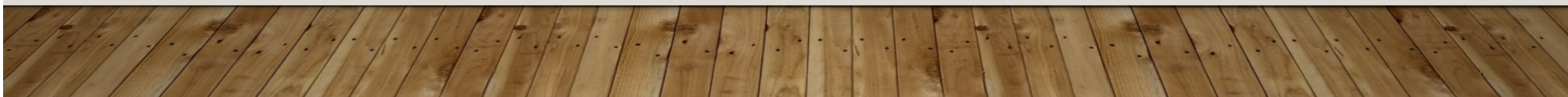
- Сначала выполнится эта строчка (на приведенном примере она вывела на экран строку "Hello world"), а затем обязательно появится строка, начинающаяся с символов **=>**.
- Эта строка — результат, который вернуло введенное вами выражение.

ПРИНЦИП РАБОТЫ

- Выражением может быть любое корректное выражение на языке Ruby. Если выражение некорректное, вы просто получите в консоли ошибку. Так же, как вы бы получили ее при обычном запуске программы. Например, очень удобно использовать Ruby консоль в качестве продвинутого калькулятора (скажем, считать задолженность по оплате электроэнергии):

```
3.0.1 :002 > 3.91*(2342-2291) + 3.23*(1745-1622)
=> 596.7
3.0.1 :003 >
```

- Как видите в этом случае после вашей строки, выводится только строка с результатом выполнения выражения. И затем IRB снова ждет ваших распоряжений.



МНОГОСТРОЧНЫЙ КОД

- Если вам нужно выполнить в консоли код, который не умещается в одну строку, можете заключить его внутри блока **begin** — **end**.
- После ввода **begin**, консоль будет запрашивать все новые и новые строки, пока не встретит конец этого блока **end**. И тогда только разом выполнит весь этот блок кода.

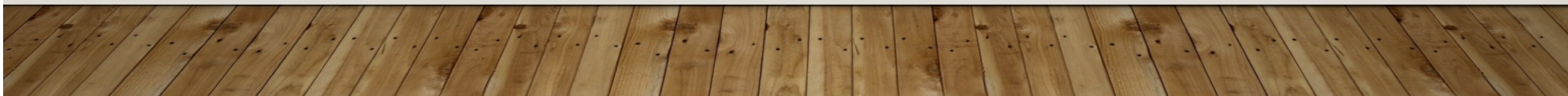
```
3.0.1 :003 > begin
3.0.1 :004 >     sleep 3
3.0.1 :005?> puts ['orel', 'reshka'].sample # вывести орел или решку
3.0.1 :006?> end
orel
=> nil
3.0.1 :007 >
```

МНОГОСТРОЧНЫЙ КОД

- Надпись `orel` и `=> nil` появятся на экране только через 3 секунды после ввода 6-й строки. Кстати, комментарии тоже можно писать (любой руби код), но зачем? ;)
- Для выхода из консоли достаточно набрать волшебное слово `exit` и вернуться к нормальной жизни, найти работу, завести семью...

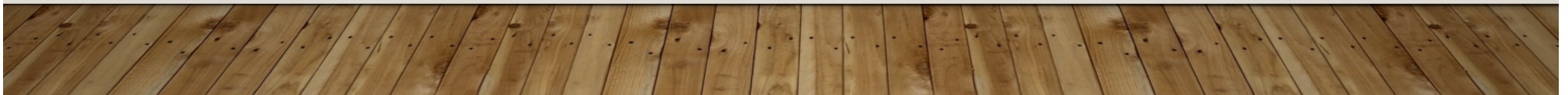
КАК И КОГДА ПОЛЬЗОВАТЬСЯ IRB

- В консоль при желании можно скармливать огромные программы. Но делать так не стоит. Консоль — это фастфуд, когда нужно срочно выполнить немножко какого-то рубли кода.
- Когда она актуальна:
 - Когда нужен продвинутый калькулятор :)
 - Если нужно быстро написать очень простенькую **одноразовую** (!) программу
 - Если нужно проверить как работает какой-то кусочек кода. Например, вы не уверены в каком-то новом методе, а документация не прояснила ситуацию
 - Чтобы наглядно увидеть и еще лучше понять как работает интерпретатор Ruby (сверху вниз, строчка за строчкой).



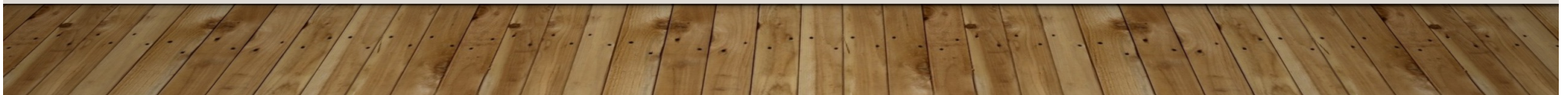
ПАЛИНДРОМ С ПРОБЕЛАМИ

- Сделайте вторую версию программы «Палиндром», которая не учитывает наличие пробелов/знаков препинания во фразе.
- Сперва с помощью `irb` поймите, как убрать из строки все пробелы (не только те, что по краям) и знаки препинания. Посмотреть все методы объекта можно вызвав у него метод `methods`.
- Если не выйдет найти метод с помощью `irb`, погуглите или посмотрите документацию на класс `String`.
- Проверьте в консоли как работает метод `delete` у строки. Попробуйте вызвать его с разными аргументами.



ПАЛИНДРОМ С ПРОБЕЛАМИ

- Сделайте вторую версию программы «Палиндром», которая не учитывает наличие пробелов/знаков препинания во фразе.
- Сперва с помощью `irb` поймите, как убрать из строки все пробелы (не только те, что по краям) и знаки препинания. Посмотреть все методы объекта можно вызвав у него метод `methods`.
- Если не выйдет найти метод с помощью `irb`, погуглите или посмотрите документацию на класс `String`.
- Проверьте в консоли как работает метод `delete` у строки. Попробуйте вызвать его с разными аргументами.



СПРАВОЧНАЯ ИНФОРМАЦИЯ

- [Про блоки кода begin ... end \(eng\)](#)
- [Официальная документация irb \(eng\)](#)
- [Interactive Ruby Shell \(вики\)](#)

СПАСИБО ЗА ВНИМАНИЕ!

Интерактивная консоль Ruby