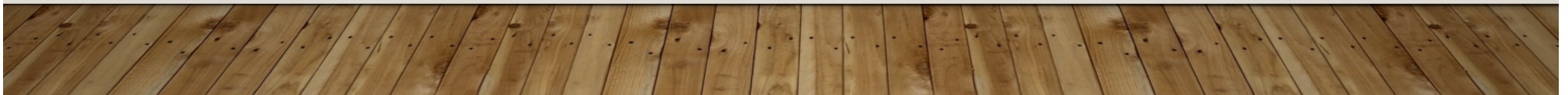


ЛЕКЦИЯ 15

БИБЛИОТЕКИ И RUBYGEMS

ПЛАН ЗАНЯТИЯ

1. Использование библиотек — зачем, почему и как
 2. Система библиотек Ruby Gems
 3. Пишем свою программу для отправки электронной почты
- Мы научимся работать с библиотеками, узнаем, как пользоваться утилитой **gem** и её командами: **gem update**, **gem list**, **gem install**. Как находить и устанавливать нужные библиотеки.
 - Узнаем, как с помощью **require** добавлять в программу установленные библиотеки и напишем простенькую программу для отправки почты с помощью гема «Pony».



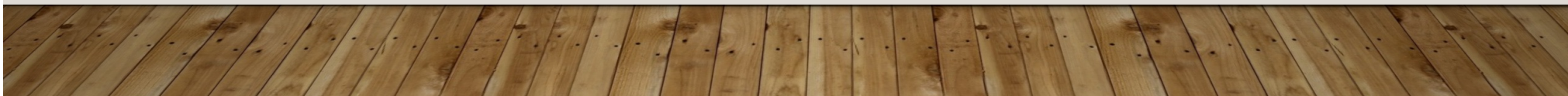
ЧТО ТАКОЕ БИБЛИОТЕКИ В ПРОГРАММИРОВАНИИ?

- Программисты — очень прагматичные и ленивые люди, они ненавидят писать одно и то же в одной и той же программе несколько раз. Поэтому они придумали циклы и методы, потом они решили, что они не хотят писать одно и то же в разных программах и придумали классы и подключаемые файлы.



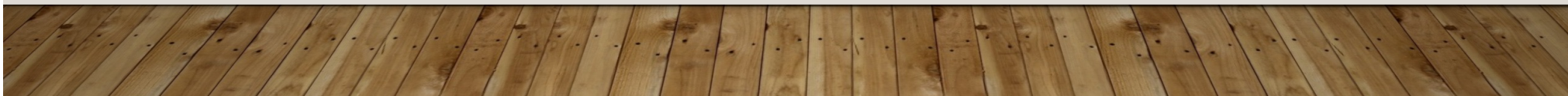
ЧТО ТАКОЕ БИБЛИОТЕКИ В ПРОГРАММИРОВАНИИ?

- Через какое-то время, эти твари обленились окончательно и начали заимствовать код друг у друга: так появились библиотеки.
- Библиотека — это набор программ, классов, методов и других файлов, который служит для решения какой-то задачи. И предназначенный для того, чтобы использоваться внутри других программ. Например, Windows это не библиотека (хотя тоже набор программ и классов по сути) — потому что не предназначена для использования внутри других программ, это самостоятельная и независимая программа.



ЧТО ТАКОЕ БИБЛИОТЕКИ В ПРОГРАММИРОВАНИИ?

- А вот скажем, библиотека [Pony](#) для отправки электронной почты, которая позволяет написать свою программу на Ruby для отправки e-mail, это другое дело — она предназначена для использования в программах.
- Библиотеки как правило пишут уже очень крутые разработчики, однако пользоваться библиотеками можно и нужно всем, даже самым полным новичкам.
- Это во-первых, сокращает время на написание программ, во-вторых, вам не нужно знать до конца, как работает та или иная библиотека, чтобы её использовать.



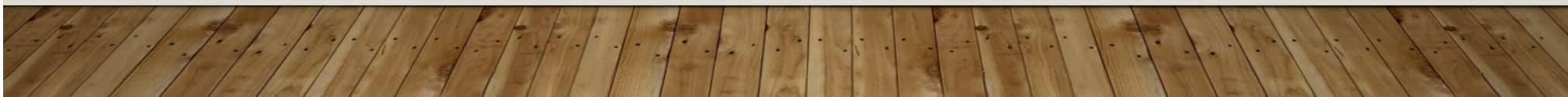
ЧТО ТАКОЕ БИБЛИОТЕКИ В ПРОГРАММИРОВАНИИ?

- Использование библиотеки для написания программ — как использование микроволновки для приготовления еды: вы не понимаете, как она работает внутри, но через какое-то время уже не можете без неё жить, здорово сокращая время процесса.



ЧТО ТАКОЕ БИБЛИОТЕКИ В ПРОГРАММИРОВАНИИ?

- Библиотеки придумали очень давно, поэтому сейчас для любой цели есть своя библиотека в любом языке программирования. И не одна. Библиотеки для работы с файлами, для работы с сайтами, с базами данных, с картинками, с музыкой и так далее.



СИСТЕМА УПРАВЛЕНИЯ БИБЛИОТЕКАМИ RUBYGEMS

- Когда разработчик пишет библиотеку и она лежит только на его компьютере, от этого никому ни холодно, ни жарко. Чтобы эффективно обмениваться этими библиотеками, необходимо было создать центральное хранилище этих библиотек, как такой научный городок, в котором только библиотеки. Каждый, кому нужна какая-то книга, знает, куда ему ехать.



СИСТЕМА УПРАВЛЕНИЯ БИБЛИОТЕКАМИ RUBYGEMS

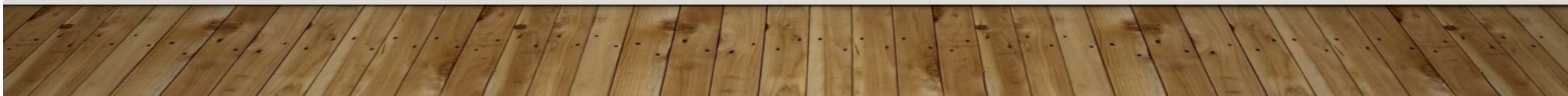
- В Ruby такая система работы с библиотеками называется **RubyGems**. Она устанавливается вместе с Ruby.
- Проверьте это, выполнив в консоли команду:

```
gem -v
```

- Чтобы обновить эту утилиту до последней версии, просто напишите:

```
gem update --system
```

- **RubyGems** сама себя скачает и установит, написав вам, когда закончит этот.



УСТАНОВКА БИБЛИОТЕК

- Теперь немного о хранении библиотек: они все лежат на далёком-далёком сайте и занимают там немало места: сама утилита **gem** весит сущие килобайты, а вот всё, что она может скачать, измеряется, наверное, терабайтами.
- Именно поэтому для использования библиотеки её сперва необходимо скачать и установить на ваш локальный компьютер.

УСТАНОВКА БИБЛИОТЕК

- Список того, что скачалось и установилось вместе с ruby, вы можете посмотреть, набрав в консоли:

```
gem list
```

- А чтобы скачать и поставить новую библиотеку, нужно знать, как она называется.
- Например, чтобы установить библиотеку `unicode_utils`, которая здорово помогла бы нам в наших выкрутасах с кодировкой в Windows, нужно использовать команду `gem install`, после которой написать название библиотеки:

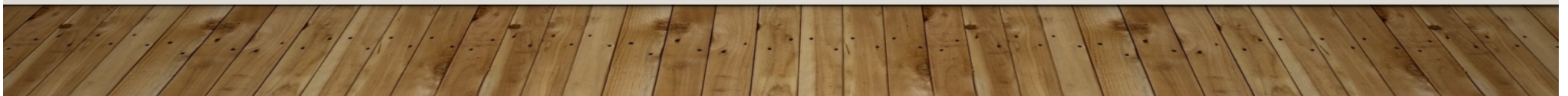
```
gem install unicode_utils
```

ПОИСК БИБЛИОТЕК

- Где искать библиотеки? Если вкратце, то в гугле.
- Все библиотеки хранятся на сайте rubygems.org, но искать их удобнее через сайт ruby-toolbox.com, где самые популярные Ruby-библиотеки описаны и рассортированы по категориям.
- Искать библиотеки можно и на google.com. Просто наберите в поисковой строке гугла:

```
ruby gem unicode
```

- И он выдаст вам несколько ссылок на различные страницы сайтов (rubygems.org обычно), где вы сможете найти нужную вам библиотеку.



БИБЛИОТЕКА PONY ДЛЯ ОТПРАВКИ ПОЧТЫ

- Поставим задачу на сегодня — **написать программу, которая будет отправлять e-mail прямо из консоли.**
- Для отправки почты в Ruby существует масса различных по сложности и возможностям библиотек. Мы выберем простую — **pony**.
- Для того, чтобы её установить, как мы уже писали, нужно написать в консоли (в какой угодно папке) команду **gem install**:

```
gem install pony
```


ПОДКЛЮЧЕНИЕ БИБЛИОТЕК В ПРОГРАММАХ

- Вы купили микроволновку и поставили куда-нибудь на кухне. Но греть еду в ней всё равно пока не можете. Её нужно подключить.
- Также и в программировании: чтобы методы и классы библиотеки были доступны в вашей программе, вам необходимо сперва её подключить.
- Это делается уже знакомым нам методом `require` (кстати, от английского "запросить, потребовать").
- На этот раз в качестве параметра методу `require` нужно передать не строку с путём к файлу, а строку с названием библиотеки:

```
require "pony"
```

ПИШЕМ ПРОГРАММУ ДЛЯ ОТПРАВКИ ПОЧТЫ

- Задачу мы поставили пару абзацев выше по тексту, приступим к делу. Сразу отметим, что пароль от почты программа также будет спрашивать у пользователя.
- **Никогда не храните пароли в текстах программ**, чтобы злоумышленники, получив каким-то образом доступ к вашей программе, не могли слать с её помощью письма от вашего имени. Так-то.
- Создадим нашу программу для отправки почты **send_mail.rb** в нашей рабочей папке 15-го урока: `c:\%username\lesson15` и напишем в ней код отправки почты.

ПИШЕМ ПРОГРАММУ ДЛЯ ОТПРАВКИ ПОЧТЫ

- Сперва мы сохраним в переменную `my_mail` наш адрес почты.
- Вам необходимо сюда написать адрес вашей электронной почты — тот адрес с которого вы будете слать письма:

```
my_mail = my_mail@mail.ru
```

ПИШЕМ ПРОГРАММУ ДЛЯ ОТПРАВКИ ПОЧТЫ

- После этого мы спрашиваем уже знакомой нам конструкцией `gets` у пользователя все нужные нам поля: пароль от почты, кому слать письмо и текст самого письма. Сохраняем каждое поле в отдельную переменную.

```
puts "Введите пароль от вашей почты #{my_mail} для отправки письма:"  
password = STDIN.gets.chomp  
  
puts "Кому отправить письмо? Введите адрес:"  
send_to = STDIN.gets.chomp  
  
puts "Что написать в письме?"  
body = STDIN.gets.chomp.encode("UTF-8")>
```

ПИШЕМ ПРОГРАММУ ДЛЯ ОТПРАВКИ ПОЧТЫ

- А потом отправляем почту:

```
Pony.mail({  
  :subject => "Привет из программы на руби!",  
  :body => body,  
  :to => send_to,  
  :from => my_mail,  
  :via => :smtp,  
  :via_options => {  
    :address => 'smtp.mail.ru',  
    :port => '465',  
    :tls => true,  
    :user_name => my_mail,  
    :password => password,  
    :authentication => :plain  
  }  
})
```


ПИШЕМ ПРОГРАММУ ДЛЯ ОТПРАВКИ ПОЧТЫ

- Обратите внимание, что для отправки почты мы вызвали метод `mail` нового класса `Pony`, который подключился с помощью `require 'pony'`.
- В качестве параметров этому методу передаётся ассоциативный массив, что это такое мы узнаем в следующих лекциях, пока просто скопируйте это себе в программу.
- Также важно понимать, что настройки, которые указаны после `via_options` работают только для отправки писем с помощью сервера `**mail.ru**`, поэтому если у вас почта у другого провайдера, вам нужно заменить этот блок другим кодом.

СКРЫТИЕ ПАРОЛЯ ПРИ ВВОДЕ

- Если оставить всё, как есть, то при вводе пароля пользователь засветит его любому, кто в этот момент будет смотреть на его монитор.
- Это не здорово. Не безопасно.
- Поэтому мы скроем пароль от посторонних глаз при вводе. Делается это с помощью другой библиотеки `io/console`.
- Эта библиотека устанавливается вместе с Ruby, но все равно надо подключить ее в программе методом `require`

```
require 'io/console'
```

СКРЫТИЕ ПАРОЛЯ ПРИ ВВОДЕ

- После этого нам станет доступен метод `noecho`, в который можно будет передать в качестве параметра наш метод `gets`, в несколько странном виде.
- Не будем сейчас подробно останавливаться на том, как это работает, просто скажем, что теперь нужно написать вместо обычного `gets` вот так:

```
password = STDIN.noecho(&:gets).chomp
```

- Теперь при вводе пароля в консоли не будет ничего отображаться. Но символы при этом будут запоминаться, поэтому будьте внимательны и не ошибитесь при вводе пароля!

ЗАПУСК ПРОГРАММЫ SEND_MAIL.RB

- Наконец, пора запускать нашу программу. Для этого как обычно в консоли переходим в нашу папку и запускаем программу:

```
cd c:\%username\lesson15  
ruby send_mail.rb
```

- Программа попросит вас ввести пароль от вашей почты: обратите внимание, вы печатаете что-то, а в консоли ничего не появляется. Не пугайтесь, набирайте в слепую - не очень удобно, зато безопасно.

ЗАПУСК ПРОГРАММЫ SEND_MAIL.RB

- Набирайте адрес, текст письма и если вы все сделали правильно и с интернетом у вас порядок — программа сразу отошлет письмо адресату. Можете проверить, послав письмо самим себе.
- Итак, в этом уроке мы научились работать с библиотеками, узнали, как пользоваться утилитой **gem** и ее командами **gem update**, **gem list**, **gem install**.
- Узнали, как с помощью **require** добавлять в программу установленные у вас в системе библиотеки и написали простенькую программу для отправки почты с помощью библиотеки «Pony».

МАТЕМАТИЧЕСКИЕ КОНСТАНТЫ

- В Ruby есть встроенная библиотека **Math**, которая помогает совершать всякие полезные математические операции, которые нам рассказывали ещё в школе: синусы, косинусы.
- С помощью модуля **Math** выведите на экран значения двух фундаментальных констант: длины окружности к её диаметру **pi** и основание натурального логарифма **e**, а после этого убедитесь, что сумма синуса + косинуса угла **pi** равны -1 (выведите сумму на экран).

МАТЕМАТИЧЕСКИЕ КОНСТАНТЫ. ПОДСКАЗКА

- Почитайте документацию модуля **Math**:
- <https://ruby-doc.org/core-2.4.0/Math.html>
- Для того, чтобы использовать константы модуля **Math** нужно набирать их большими буквами после названия модуля через двойное двоеточие:
- Чтобы использовать методы модуля **Math**, нужно набирать их маленькими буквами через точку:
- Чтобы округлить сумму синуса и косинуса, используйте метод **round(2)**.

```
Math::PI  
Math::E
```

```
Math.sin  
Math.cos
```

ТРАНСЛИТЕРАТОР

- Напишите программу-транслитератор: программу, которая берёт русский текст и переводит его в созвучный английский. А если пользователь ввел фразу на английском, программа должна транслитерировать на русский.

- **Например:**

Введите фразу для транслитерации:

Отличный ёж

Otlichnyy yezh

- **или:**

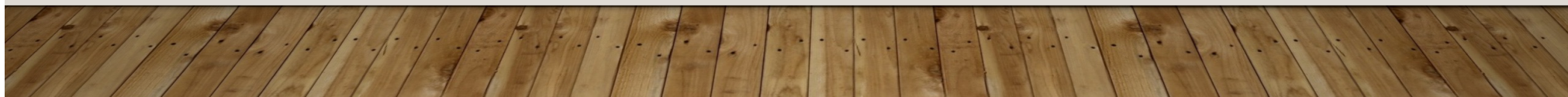
Введите фразу для обратной транслитерации:

Мама mila ramu

Мама мила раму

ТРАНСЛИТЕРАТОР. ПОДСКАЗКА

- Конечно, Вам понадобится для этого какая-то библиотека. Вот Вам некоторые:
- <https://github.com/tjbladez/translit>
- <https://github.com/dalibor/cyrillizer>
- <https://github.com/yaroslav/russian>
- https://github.com/torbjon/translit_net
- Сами разберитесь, как ими пользоваться. Чтобы проверить, на каком языке фраза можно, например, сравнить её с транслитерированным вариантом.
- Например, если при транслитерации в английский ничего не поменялось — фраза на английском.



ОТПРАВКА ПОЧТЫ С ТЕМОЙ

- Сделайте так, чтобы в нашей программе `send_mail.rb` тема письма также вводилась пользователем из консоли при отправке.

ОТПРАВКА ПОЧТЫ С ТЕМОЙ. ПОДСКАЗКА

- Попросите пользователя кроме пароля, адреса отправки и тела письма указать также и тему письма, сохраните её в переменную `subject`.
- Укажите эту переменную `Pony` вместо стандартной темы в параметре `subject`:

```
Pony.mail(  
  subject: subject,  
  body: body,  
  ...
```

СПРАВОЧНАЯ ИНФОРМАЦИЯ

- [Решение проблемы с SSL на Windows - 1](#)
- [Решение проблемы с SSL на Windows - 2](#)
- [База популярных Ruby-библиотек с описаниями](#)
- [О библиотеках в Ruby, Gems \(1\)](#)
- [О библиотеках в Ruby, Gems \(2\)](#)
- [Как не светить пароль](#)
- [Библиотека Pony](#)
- <https://ruby-doc.org/core-2.4.0/Math.html>

СПАСИБО ЗА ВНИМАНИЕ!

Библиотеки и RubyGems