



# I/O and Files

by Liah Hansen and Sarah Allen

# IO: Input/Output



- IO class is the basis for all input and output in Ruby
- IO objects represent readable and/or writable connections to disk files, keyboards & screens
- File is a subclass of IO and allows for reading and writing files in ruby

# common modes for opening I/O ports



- "r" Read-only, starts at beginning of file (default)
- "r+" Read-write, starts at beginning of file
- "w" Write-only, truncates existing file to zero length or creates a new file for writing
- "w+" Read-write, truncates existing file to zero length or creates a new file for reading and writing

# open a file



- Creating the file object opens the file:

```
>> f = File.new("abc.rb")
```

```
=> #<File:abc.rb>
```

# open file – with a block

- `File.open` is synonymous with `File.new`, except that it can take a block:

```
>> File.open("abc.rb", "r+") do |f|  
      f << "hello"  
    end  
=> #<File:abc.rb (closed)>
```

# read a file

- return a single line

```
>> f.gets  
=> "aaa\n"
```

- return the whole file after current position

```
>> f.read  
=> "aaa\nbbb\nccc"
```

- returns file as an array of lines

```
>> f.readlines  
=> ["aaa\n", "bbb\n", "ccc"]
```

# position & rewind

```
>> f.rewind
```

```
=> 0
```

```
>> f.pos
```

```
=> 0
```

```
>> f.gets
```

```
=> "aaa\n"
```

```
>> f.pos
```

```
=> 4
```

# iterate over lines



- File objects are enumerable:

```
>> f.each {|line| puts line}
```

```
aaa
```

```
bbb
```

```
ccc
```



# use any enumerable method

```
>> f.map {|line| line.chomp + "... "}
=> ["aaa... ", "bbb... ", "ccc... "]
```

```
>> f.any? {|line| line =~ /aaa\n/ }
=> false
```

```
>> f.rewind
=> 0
```

```
>> f.any? {|line| line =~ /aaa\n/ }
=> true
```

# write to a file

## □ “w+” argument

```
>> my_file = File.new("data.txt", "w+")
```

```
=> #<File:def.rb>
```

```
>> my_file.write("something")
```

```
=> 9
```

```
>> my_file.rewind
```

```
=> 0
```

```
>> my_file.read
```

```
=> "something"
```

# directories



```
>> d = Dir.new("/Users/liahhsen/s/dir")  
=> #<Dir:/Users/liahhsen/s/dir>
```

```
>> d.entries  
=> [".", "..", "abc.rb", "another_subdir",  
    "def.rb", "ghi.rb", "subdirectory"]
```

# count files in directory

---

```
>> i=0
```

```
=> 0
```

```
>> d.each {|x| i+=1; puts i}
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

# iterate over files in directory



```
>> ruby_files = Dir["*.rb"]  
=> ["abc.rb", "def.rb", "ghi.rb"]
```

```
>> ruby_files.map do |file|  
    "rb_" + file  
end  
=> ["rb_abc.rb", "rb_def.rb",  
    "rb_ghi.rb"]
```