

# Working with Files

*CS105 : Saelee*

So far, our programs either  
use **static** data or read  
input **from the user**

Sometimes, this isn't  
enough ...

# Software that uses Files

- Word processors (e.g., doc files)
- Media players (e.g., mp3, mov, wav files)
- Text editors (e.g., text, source code files)
- Spreadsheet programs (e.g., Excel, CSV files)
- Games (e.g., texture, level description, save files)
- ... and pretty much every other non-trivial piece of software!

# Types of Files

- “Text” files
  - Contents are human legible
  - Typically makes use of “ASCII” encoding to map from bit-sequences to characters
- “Binary” files
  - Contents are **not** generally human legible
    - Sequences of 0s and 1s
  - Meant to be parsed by software or hardware

# Working with files in Ruby

- The `File` class
- Methods include:
  - `open` for opening / creating a file
  - `close` for closing a file
  - `puts` and `gets` for writing and reading a line
  - `read` for reading in the entire contents of a file
  - `each_line` for going through the lines in a file

# File.open

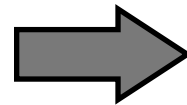
- Takes two parameters:
  - The filename, or path to a file, e.g., “foo.txt”
  - The mode in which to open the file:

Mode	Meaning
r	Read-only, start at beginning of file
r+	Read & write, start at beginning of file
w	Write only; always start with an empty file
w+	Read & write; always start with an empty file
a	Write only; start at end of file if it already exists
a+	Read & write; start at end of file if it already exists

# A File Object

- `File.open` returns a **File object** that we can use for reading and writing
  - Have to close the file after using it (to make sure it's properly saved)

```
f = File.open("test.txt", "w")  
f.puts("Hello, file!")  
f.close
```



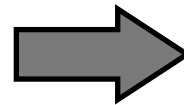
test.txt

```
Hello, file!
```

# A Better Way...

- It's important to always close a file after using it
  - Problem: we can forget to call close
- If we call open with a block, we can use the file within the block, and Ruby will close it for us when the block exits

```
File.open("test.txt", "w") do |f|  
  f.puts("Hello, file!")  
end
```



test.txt

```
Hello, file!
```