



---

## REVISION

---

- Simple Data Types
  - Number, Float, String, Boolean, DateTime
- Operators
  - Primitive, Boolean & Relational
- Variable assignment
- Conditional Statements
  - if, elsif & else
- Methods

---

## REVISION

---

# EXPRESSIONS

▸  $2 + 3$

▸  $6 - 3$

▸  $4.5 * 3.5$

▸  $6 / 3$

---

## REVISION

---

# ASSIGNMENT / EXERCISE

- A variable has a name and a value
  - $x = 2$
  - $y = 3$
  - $z = x * y$
- Variables can be used in statements
  - `puts x + ' * ' + y + ' = ' + z`

---

## REVISION

---

# CONDITIONAL STATEMENTS

- Expression evaluates
  - true (if, elsif)
  - false (else)

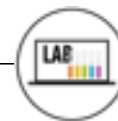
---

## REVISION

---

# METHODS

- How to define a method
  - Keyword def
  - Unique method name
  - 0 or more parameters
  - Only 1 return value
- How to invoke/call a method



# Exercise

- › Clone the repository `shirren/bewd4_exercises`
- › Complete the code in `class3/bewd_math.rb`
  - › 15 min time limit

---

**FUNDAMENTALS**

---

# **COLLECTIONS**



---

## FUNDAMENTALS

---

# ARRAYS

---

## COLLECTIONS

---

# ARRAYS

- Sequence of values
  - Like a pill box
  - Each position has a label
  - The label is always a number
    - Starts at 0!
- Ordered
  - Ruby arrays can store multiple types





# Arrays

- Supports additional methods
  - reverse
  - friendly names to access data
    - first
    - last
- Lets just dive in!

# **HASHES**

---

## COLLECTIONS

---

# HASH

- Collection of values
  - Like a pill box
  - Each position has a label
  - The label can vary
    - Be a number
    - Be a string
    - Be a symbol
- Label is referred to as the key





# Hashes

- Supports additional methods
  - `has_key?`
    - `key?`
    - `include?`
  - `has_value?`
  - `keys`
  - `values`
- Lets just dive in!

---

## FUNDAMENTALS

---

# SYMBOLS

---

## SYMBOLS

---

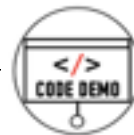
- Special type in ruby, used extensively
  - Always preceded by a colon
  - Cannot contain spaces or numbers
- Are used because:
  - Use less memory
  - Easier to compare to other objects
  - Cleaner in syntax
- Examples:
  - :hello, :this\_is\_a\_symbol



## FUNDAMENTALS

---

**PRIMARYLY USED AS  
KEYS FOR HASHES**



# Hash rocket syntax

- Ruby 1.8 & <
  - `:key => value`
- Ruby 1.9+
  - `key: value`
- Ruby developers prefer the newer syntax
  - Remnants of old syntax still around

---

## FUNDAMENTALS

---

# REPETITION



# Repetition

- Loops are used to repeat lines of code
- Ruby loops are a little different:
  - `.times`
  - `.each`
  - `.map`
  - `.reduce`
- Readup on `.upto` and `.downto`
- Ruby has `for`, `while` and `until`

---

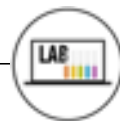
**5-10 MIN BREAK**

---



**TREAT!**

<http://www.sitepoint.com/history-ruby/>



# Exercise

- Fizz Buzz

- TIME : 20 min

- Write a program that prints numbers from 1 to 100.
    - But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”.
    - For numbers which are multiples of both three and five print “FizzBuzz”.

**RECAP: ANY  
QUESTIONS**

# EXERCISES FROM CLASS 2



# HOW DO WE REUSE CODE?

---

**FUNDAMENTALS**

---

**METHODS**

---

## FUNDAMENTALS

---

# GROUP METHODS?

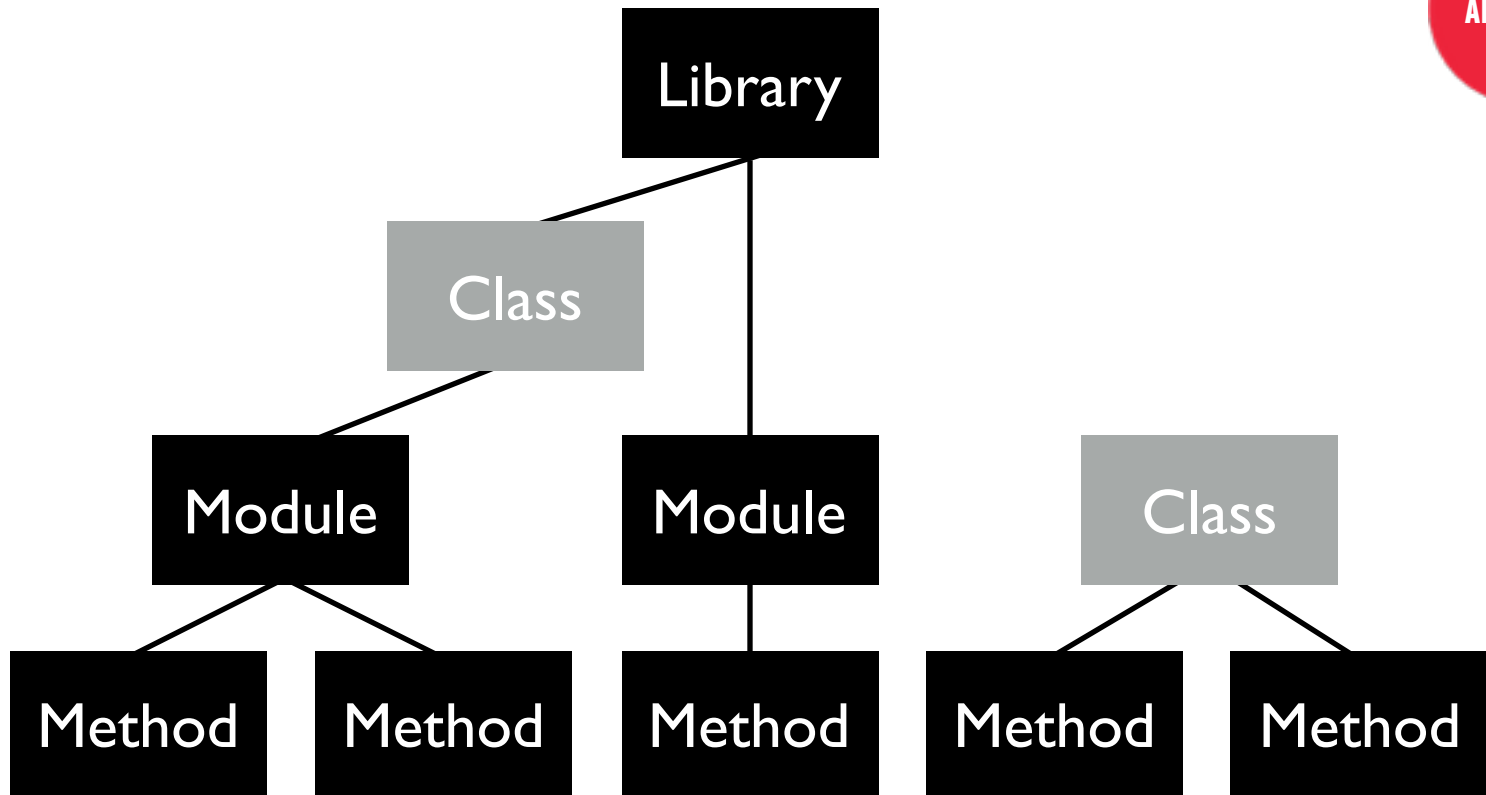
**MODULES  
ARE POWERFUL!**



# Modules

- Used to group methods
- Modules can be re-used
  - We will see how in class 5
- Keyword module
  - Module name must be unique

# **HOW DO WE REUSE MODULES?**





# Libraries

- Also known as Gems
- Name some famous Gems
  - Rails, Devise, Capybara etc
  - So many ...
- How do we use libraries?
  - 2 techniques
    1. Install locally
    2. Use an API





[GEMS](#)[GUIDES](#)[CONTRIBUTE](#)[SIGN IN](#)[SIGN UP](#)

# Find, install, and publish RubyGems.

**5,208,877,317**

DOWNLOADS &amp; COUNTING

[Install RubyGems](#) STATUS UPTIME

RubyGems.org is the Ruby community's gem hosting service. Instantly publish your gems and install them. Use the API to interact and find out more information about available gems. Become a contributor and enhance the site with your own changes.

[STATUS](#)[UPTIME](#)[CODE](#)[DISCUSS](#)[STATS](#)[BLOG](#)

---

## FUNDAMENTALS

---

# Libraries cont...

- Need to use many libraries when building a Rails application
  - Use Bundler for installing Gems
    - Instead of gem install ...
- Libraries have version numbers
  - More on this when we start with Rails ...

# **HOMEWORK!**