

# Lesson 7.03: Methods ## Learning Objectives Students will be able to... \* Define and identify: **\*\*method\*\***, **\*\*`\_\_str\_\_`\*\***, **\*\*`\_\_add\_\_`\*\***, **\*\*operator overloading\*\*** \* Create a class with an **`\_\_init\_\_`** method \* Understand and use the **`self`** argument \* Instantiate a class with an argument ## Materials/Preparation \* [Do Now] \* [Lab - Kangaroo Class] ([printable lab document]) ([editable lab document]) \* [Associated Reading - section 7.3](https://tealsk12.gitbook.io/intro-cs-2/readings#7-3) \* Read through the do now, lesson, and lab so that you are familiar with the requirements and can assist students ## Pacing Guide | **\*\*Duration\*\*** | **\*\*Description\*\*** | | ----- | ----- | 5 Minutes | Do Now | | 10 Minutes | Lesson | | 35 Minutes | Lab | | 5 Minutes | Debrief | ## Instructor's Notes #### 1. Do Now \* Display the Do Now on the board. \* Students will find that when they try to print the two different Time objects, it produces output that's not particularly useful or readable. \* Students will also discover that adding objects doesn't work...yet! #### 2. Lesson ##### Instruction - Method \* a function inside of a class. \* The first argument is always **`self`**. ##### Discussion \* Ask students what method we have already seen and used previously. (**`init`**) \* Ask students how they would distinguish between the two time variables. ##### Instruction - **`\_\_str\_\_`** \* Need a method called **`\_\_str\_\_`**. \* This will get called when you print an object \* it returns a string that is easy to read and understand ##### Activity \* Have the students practice writing **`\_\_str\_\_`** for the **`Time`** class for 5 minutes. \* Have a student write up their string method on the board. ##### Instruction **`\_\_add\_\_`** \* A method that gets called when the plus sign is used between two **`Time`** objects. \* In this case it takes as parameters **`self`** and another **`Time`** object and returns a **`Time`** object that is the sum of both. \* Overwriting add is called **\*\*operator overloading\*\*** because you are re-writing the code used to make the **+** work. ##### Demonstration \* Work together with students to come up with the add time algorithm. #### 3. Lab \* Have students finish up the time adding method. \* Have students work on kangaroo lab. #### 4. Debrief \* Go over students' questions and demonstrate some students' successfully completed labs. \* Review what a method is, as well as what specific methods were used in today's lab. ## Accommodation/Differentiation Students that are moving quickly should work on the bonus assignment in the lab or assist a partner that is struggling. ## Forum discussion [Lesson 7.03: Methods (TEALS Discourse Account Required)](https://forums.tealsk12.org/c/2nd-semester-unit-7-classes/lesson-7-03-methods) [Do Now]:do\_now.md.html [Lab - Kangaroo Class]:lab.md.html [printable lab document]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/7\_unit/03\_lesson/lab.pdf [editable lab document]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/7\_unit/03\_lesson/lab.docx