



JuniorDevSG Coding Dojo

Session 1



What is a Code Kata?



Code Kata

A kata is an exercise in karate where you repeat a form many, many times, making little improvements in each.

Source: <http://codekata.com/>



**WAX ON,
WAX OFF.**

Code Kata

- You need to try it as many times as it takes, and be comfortable making mistakes.
- You need to look for feedback each time so you can work to improve.
- Remember that the point of the kata is not arriving at a correct answer. The point is the stuff you learn along the way. The goal is the practice, not the solution.

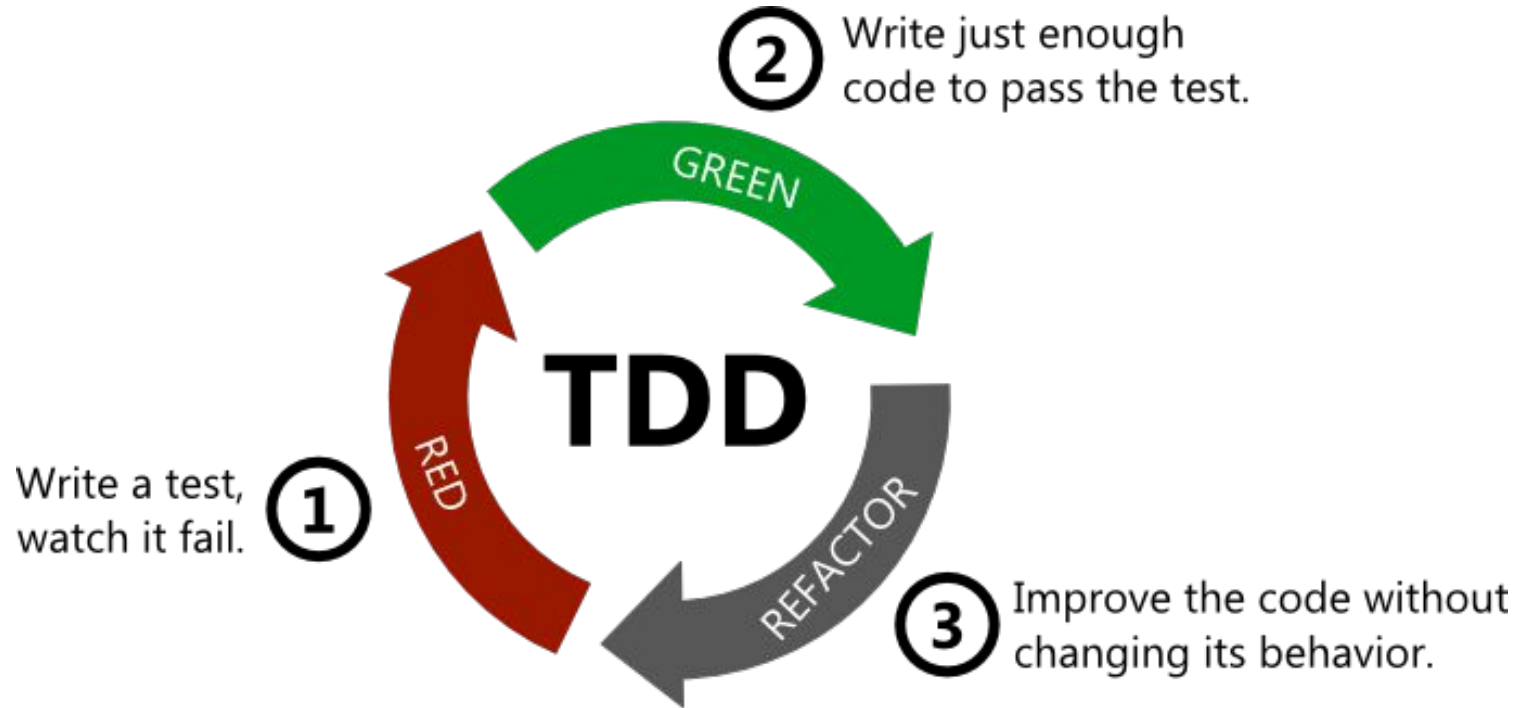


**WAX ON,
WAX OFF.**

Test Driven Development



How?



TDD Rules

1. **Start with a test:** only start writing production code when your automated test has failed.
2. **Baby steps:** Only 1 failing test at a time.
3. **Baby steps:** Write small test.
4. Write enough production code to make the failing test pass. No extra code.



TDD Rules

5. **Baby steps:** Implement simplest algorithm first, then generate it later when you identify some patterns.
6. Don't forget about refactoring.
7. Refactor your tests too!
8. Don't refactor when your tests are failing. Make tests pass first.



Pair Programming





What is Pair-Programming?

- An agile software development technique in which two programmers work together at one workstation (1 computer, 2 sets of keyboard & mouse, 2 monitors).
- Roles:
 - **Driver** - writes code
 - **Observer / Navigator** - reviews each line of code as it is typed in.
- The two programmers switch roles frequently.





Practical Tips

- **Be conversational.** Talk through your thought process. Because people can't read minds.
- **Plan.** Take time to plan what you hope to achieve at the beginning of the pairing session.
- **Take notes.** I find that sometimes, it's faster to communicate ideas on paper/whiteboard than on the screen. I draw architectural diagrams and bullet points on paper (or on a whiteboard) to brainstorm ideas.



Practical Tips

- **Agree on a pairing workflow.** There are many pairing technique. My favourite is the Ping-Pong method - where one pair writes a test, and the other pair writes the code that passes the test, and repeat.
- **Focus on the work at hand.** Don't get distracted by non-work PM, emails and social media pings. Don't waste the other pair's time.
- **Share keyboard time.** Don't hog the keyboard. Let the other pair speak through code as well. Be patient with them.



Practical Tips

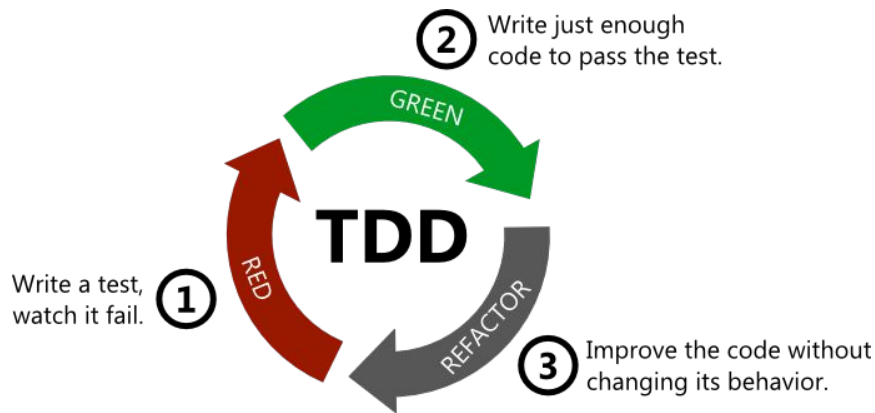
- **Use on screen cues** to help bring focus. Use line number to help zoom in on code. Learn to bring focus to things on screen with the mouse cursor instead of your finger.
- **Take breaks.** Do take breaks frequently. Once you reach a natural breakpoint, like after a Git check commit has been made, do take a short break to clear the mind. Pairing can be very intense and involves much concentration and focus - so taking frequent breaks will help rest your mind.

Today's Code Kata is...



Things to note...

- Pair-up with someone doing the same language
- Use TDD to come up with the solution
- Enjoy!





Helpful Resources

- https://github.com/JuniorDevSingapore/coding_dojo
- <https://speakerdeck.com/miccheng/how-much-should-we-test>

Developers! Find your pair!



Break Time



Show and Tell

