



# Conversational writing

*Kumar Dhanagopal, STC India annual conference*

December 11, 2015

# Here's what we'll talk about today...

- ▶ What is conversational writing?
- ▶ Why/when is it useful?
- ▶ How can we write conversationally?

# So what is conversational writing?



# What conversational writing is NOT ...

- ✗ Not new!
- ✗ Not writing as we speak
- ✗ Not necessarily about less text or smaller words
- ✗ Not about “dumbing” things down
- ✗ Not just about breaking “rules” of grammar
- ✗ Not easy

# Does the writing “style matter?”



[www.youtube.com/watch?v=Hzgzim5m7oU](http://www.youtube.com/watch?v=Hzgzim5m7oU)

# Does grammar matter?

1. Know the “rules”
2. Break them!
  - ▶ Split infinitives
  - ▶ End sentences with a preposition
  - ▶ Start with a conjunction
  - ▶ ...

# What's the first rule of good style?

**“... have something to say...”**

- Arthur Schopenhauer

## Read this text...

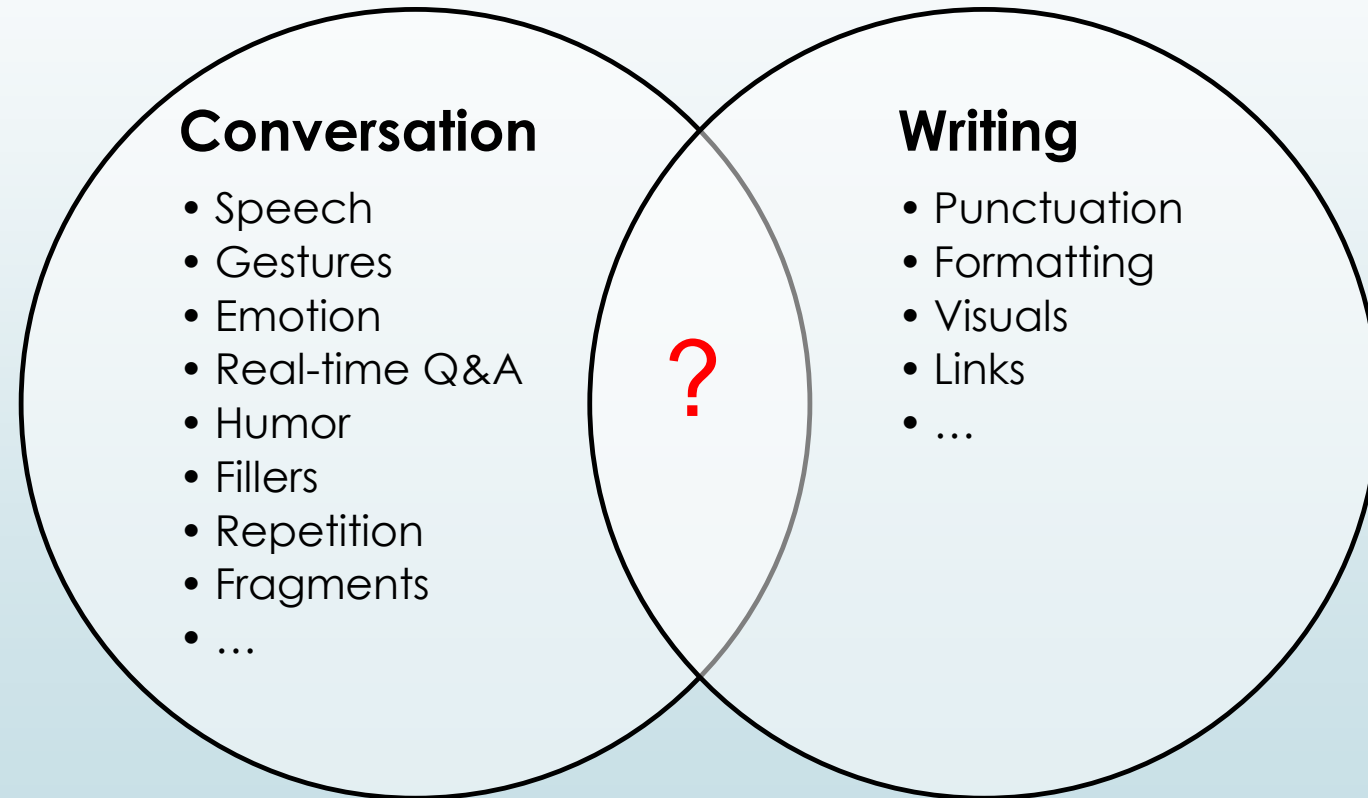
Oracle WebLogic Server is the #1 application server for developing and deploying applications across cloud environments, engineered systems, and conventional systems. Oracle WebLogic Server offers application developers modern development tooling and advanced APIs for application innovation. It provides a mission-critical cloud platform for applications requiring high performance, scalability, and reliability. Powerful, integrated management tools simplify operations and reduce management costs. Finally, Oracle WebLogic Server provides the foundation for the Oracle Fusion Middleware portfolio of products.

**How do we write this conversationally?**





# Is conversational writing even possible?



# Our relationship with our users...

Our users have the **right** to understand what we write.

We have a **duty** to write to be understood.

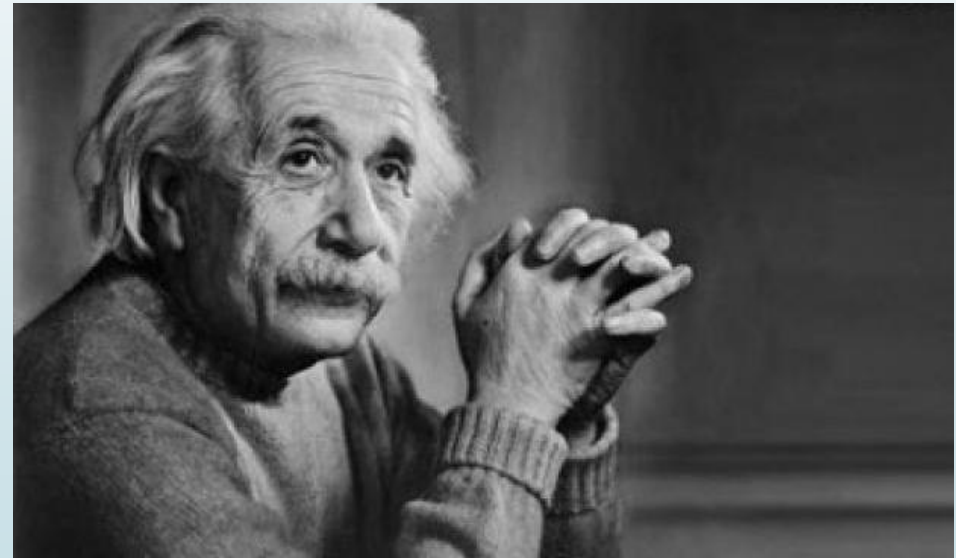
# Rule #1: Know your user...

- ✓ Start from something that your user already understands
- ✓ Focus on what's most important, to the user
- ✓ Establish credibility, early
- ✓ Respect the user

**But each user is different, right?**

## Rule #2: Know your content...

If you can't explain it simply, you don't understand it well enough.



## Read this now...

*Objects* are key to understanding object-oriented technology. Look around right now and you'll find many examples of real-world objects: your dog, your television set, your bicycle. Real-world objects share two characteristics: They all have *state* and *behavior*. Dogs have state (name, color, etc.) and behavior (barking, fetching, etc.). Bicycles also have state (current gear, current pedal cadence, ...) and behavior (changing gear, changing pedal cadence, ...). Identifying the state and behavior for real-world objects is a great way to begin thinking in terms of object-oriented programming.

Software objects are conceptually similar to real-world objects: they too consist of state and related behavior. An object stores its state in fields (variables in some programming languages) and exposes its behavior through methods (functions in some programming languages). Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication. Hiding internal state and requiring all interaction to be performed through an object's methods is known as data encapsulation — a fundamental principle of object-oriented programming.

# Focus on the core content!



The individual member of the social community often receives his information via visual, symbolic channels.



People read!

- [Richard Feynman](#)

# Make sense of abstract concepts



Disk defragmentation is the process of consolidating fragmented data on a volume (such as a hard disk or a storage device) so it will work more efficiently.



Over time, data on your hard disk gets fragmented. When opening files, your computer slows down, because it has to find and gather all of the fragmented data. You can fix this by defragmenting the disk.

- **More personal**
- **And scenario-based**

# Keep it simple

$$\begin{aligned} &1 + 2 + 3 + \dots \dots + 10 \\ &= \mathbf{10 / 2} \times \mathbf{11} \\ &= \mathbf{55} \end{aligned}$$

**How does this work?**

**1** and **10** make **11**  
**2** and **9** also make **11**  
And so on ...

We have **5** such pairs  
And **5 = 10 / 2**

So instead of adding all the numbers  
Just multiply **5** and **11**

- Carl Friedrich Gauss



# Illustrate!

## Formal

**Temperature** is a measure of the ability of a substance, or more generally of any physical system, to transfer heat energy to another physical system. The higher the temperature of an object is, the greater the tendency of that object to transfer heat. The lower the temperature of an object is, the greater the tendency of that object to be on the receiving end of the heat transfer.

## Conversational



[www.ted.com/talks/richard\\_feynman](http://www.ted.com/talks/richard_feynman)

# Get personal



This tutorial is about what type of applications you can deploy on different platforms. For each platform, an application will be designed. The goal is to learn what type of input, logic, and output can be coded for the application for it work on in each platform. Some hints are provided throughout the tutorial.



This tutorial shows **you** how to design apps for the supported platforms. **Your** goal is to learn what type of input, logic, and output **you** can code so that **your** app can work on a given platform. To help **you** along, the tutorial includes some hints.

- Use names if possible
- Ask questions

# Avoid jargon



Rate limits on “reads” from the system are defined on a per user and per application basis, while rate limits on “writes” into the system are defined solely at the user level.

Search will be rate limited at 180 queries per 15 minute window...

Source: <https://dev.twitter.com/rest/public/rate-limiting>



There's a limit to how much a user or application can read, and a limit to how much a user can write.

In a 15-minute window, you can run up to 180 search queries.

# Should you use contractions?

- ▶ Yes
- ▶ But do they work all the time?

# A few tips...

## Do what you already do...

- ✓ State the goal
- ✓ Start with a shared understanding
- ✓ Be direct
- ✓ Foresee questions
- ✓ Expect problems
- ✓ Give tips
- ✓ Warn about pitfalls
- ✓ Give feedback
- ✓ Confirm success at the end

## + a little more

- ✓ Let the words flow
- ✓ Don't worry about formatting yet
- ✓ Don't edit, at least not initially
- ✓ Read it out loud, & record yourself
- ✓ Play the recording

# Don't overdo it!



So, if you think this is, like, the storage volume that you want to delete, well then, go ahead and delete it, but before you do that, don't forget to detach it first from the instance it is attached to, as, else, you'll get an error that you can't recover from without deleting the instance and then creating it all over from scratch.



Before deleting a storage volume, detach it from the instance. Otherwise, an error will occur. To recover from the error, you'll have to delete the instance and re-create it.

# To learn more...

- *The Art of Readable Writing*, Rudolf Flesch
- <http://www.plainenglish.co.uk>
- <http://www.plainlanguage.gov/examples/index.cfm>
- <https://www.sec.gov/pdf/handbook.pdf>
- <http://faculty.georgetown.edu/tannend/TANNEN%20ARTICLES/PDFs%20of%20Tannen%20Articles/1987/Conversational%20Style.pdf>
- <https://lizprovasi.wordpress.com/2012/04/12/8-conversational-styles/>
- <http://www.aclweb.org/anthology/P80-1011>
- The Practical Guide to Writing Conversational Copy:  
<http://www.prologger.net/archives/2014/02/26/the-practical-guide-to-writing-conversational-copy/>
- Improve Your Writing with a Conversation Tone: <http://www.pickthebrain.com/blog/improve-your-writing-with-a-conversation-tone/>
- Conversational writing kicks formal writing's ass:  
[http://headrush.typepad.com/creating\\_passionate\\_users/2005/09/conversational\\_.html](http://headrush.typepad.com/creating_passionate_users/2005/09/conversational_.html)