Designing Online Learning Experiences for People

2016 June 16

Scott Murray
Learning Group
O'Reilly Media

@alignedleft
alignedleft.com
Video Training from O’Reilly
It’s like having an expert by your side.

Learning Paths
Stop hacking your way through the underbrush.
O’Reilly’s cleared the way: our Learning Paths will help you get where you want to go, whether it’s learning a programming language, developing new skills, or getting started with something entirely new.

See the learning paths
Data Visualization

VIDEO TRAINING

Successful data visualizations allow you to impart meaning and emphasis to your data points. This Learning Path will teach you how to display trends, patterns, and outliers while you discover the power of letting your data to speak. Once you’ve finished, you’ll be able to efficiently communicate volumes of data with ease.

Below are the video training courses included in this Learning Path.

1. An Introduction to d3.js: From Scattered to Scatterplot
   Presented by Scott Murray | 2 hours 52 minutes

This segment of your Learning Path will have you transforming data into visual images in no time, starting from scratch and building an interactive scatterplot by the end of the course. Learn to use d3.js, the web’s most powerful library for data visualization, to load data and translate values into SVG elements.
Live Online Training

Live online training from O'Reilly Media are a hands-on, instructor-led mix of lectures, ebook downloads, Q&A sessions, and live coding examples. Every aspect of these training courses is carefully designed to make you better at what you do.

Docker: Up and Running
Presented by Sean P. Kane
JUNE 28 & 29, 10:00AM - 12:00PM PT

Essentials of Software Architecture
Presented by Nathaniel T. Schutta, Alexander von Zitzewitz, and Mark von Richards
JULY 12, 10:00AM - 1:00PM PT
Today

Steps
11,375 steps

Resting Heart Rate
61 bpm

Recent Exercise
8 Days since you exercised.
Your Fitbit is feeling a little lethargic. Take it for a sprint.

Run
7:38 am Jun 30
4.2 miles 8'41" pace

Rowing
7:38 am Jun 22
30.00 min 268 cals

Hike
7:38 am Jun 20
45.00 min 325 cals

under calories
600 vs 800

1,674 cals
6 lbs to go
60 oz to drink
30,000 steps
Hiking Boots

Fitbit, 2015
This is the role of interface design: to facilitate learning across digital (machine) and analog (human) realms.
Information

From Wikipedia, the free encyclopedia

For the criminal process, see information (formal criminal charge).

Information (shortened as Info) is that which informs. In other words, it is the answer to a question of some kind. It is also that from which data and knowledge can be derived, as data represents values attributed to parameters, and knowledge signifies understanding of real things or abstract concepts.[1] As it regards data, the information’s existence is not necessarily coupled to an observer (it exists beyond an event horizon, for example), while in the case of knowledge, the information requires a cognitive observer.

At its most fundamental, information is any propagation of cause and effect within a system. Information is conveyed either as the content of a message or through direct or indirect observation of some thing. That which is perceived can be construed as a message in its own right, and in that sense, information is always conveyed as the content of a message.

Information can be encoded into various forms for transmission and interpretation (for example, information may be encoded into a sequence of signs, or transmitted via a sequence of signals). It can also be encrypted for safe storage and communication.

Information resolves uncertainty. The uncertainty of an event is measured by its probability of occurrence and is inversely proportional to that. The more uncertain an event, the more information is required to resolve uncertainty of that event. The bit is a typical unit of information, but other units such as the nat may be used. Example: information in one “fair” coin flip: $\log_2(2^1) = 1$ bit, and in two fair coin flips is $\log_2(4^1) = 2$ bits.

The concept that information is the message has different meanings in different contexts.[2] Thus the concept of information becomes closely related to notions of constraint, communication, control, data, form, education, knowledge, meaning, understanding, mental stimuli, pattern, perception, representation, and entropy.
Editing Information

You are not logged in. Your IP address will be publicly visible if you make any edits. If you log in or create an account, your edits will be attributed to a user name, among other benefits.

Content that violates any copyrights will be deleted. Encyclopedic content must be verifiable. Work submitted to Wikipedia can be edited, used, and redistributed—by anyone—subject to certain terms and conditions.

At its most fundamental, information is any propagandization of cause and effect within a system. Information is (Conveyed concept) (Conveyed) either as the content of a (Message) or through direct or indirect (Observation) of some (Object) (Philosophy) thing. That which is (Perception) (Perceived) can be construed as a message in its own right, and in that sense, information is always conveyed as the content of a (Message).

Information can be (Encoded) into various forms for (Transmission) (Telecommunications) (Transmission) and (Language interpretation) (Interpretation) (for example, information may be encoded into a (Sequence) of (Signs) (Semiotics) (Signals), or transmitted via a (Sequence) of ((Signal) (Electronics) Signals)). It can also be (Encrypted) for safe storage and communication.

Information resolves (Uncertainty). The uncertainty of an event is measured by its probability of occurrence and is inversely proportional to that. The more uncertain an event, the more information is required to resolve uncertainty of that event. The Bit is a typical (Unit of information), but other units such as the (Byte) (Unit) may be used. Example: information in one 'fair' coin flip: log2(sub 2 x 2) = log2(sub 2 x 2) = 2 bits.
“Uber’s Phantom Cabs”  
Alex Rosenblat, Motherboard, 2015
If...

- digital experiences are mediated by designed interfaces,
- interfaces wield power by influencing (constraining) behavior, and
- we value learning

What kinds of learning are we designing for?
Hello Processing

Casey Reas, Dan Shiffman, Scott Garner, and Scott Murray, 2013
1 `ellipse(250, 200, 200, 200);`
2 `rect(250, 200, 150, 100);`
ease()-y as Math.PI: 1,200,000ms of Fun with D3’s Animated Transitions

Scott Murray, @alignedleft, alignedleft.com

This is a presentation I gave at the Eyeo Festival in Minneapolis on June 11, 2014, adapted for the web. The talk was entirely live-coded in the JavaScript console, an experience I’ve tried to recreate here.

I recommend viewing this in Chrome, with the developer tools open. Click the next button to step through the presentation. Or, of course you can retype any of the code directly into the console yourself. Click any code block to execute it (but note that running them out of the intended order may produce unexpected results).

Hi! D3 is great for working with data, but today I’m just going to talk about transitions. The first and most important thing to know is that D3 is just manipulating the DOM. That is, D3 doesn’t “draw” anything to the screen; it simply creates new DOM elements. It also doesn’t "move" anything from one place to another; it just interpolates position values (or any other values) over time.

Let me show you what I mean. First, every action in D3 begins with a selection. Think of this as the subject of your sentence. We have to know what to act on before we can act on it.

Scott Murray, 2014
wave = function() {
  // Move to bottom
  d3.select(this)
    .transition()
    .duration(3000)
    .attr("cy", "50")
    .each("end", function() {
      // Move to top
      d3.select(this)
        .transition()
        .delay(function(d, i) {
          return i * 50;
        })
        .duration(3000)
        .attr("cy", "100");
        .each("end", wave);
    });
};

We've defined it, now let's kick things off with an initial transition, after which the wave function is called:

d3.selectAll("circle")
  .transition()
  .delay(function(d, i) {
    return i * 56;
  })
  .duration(3000)
  .attr("cy", "100")
  .each("end", wave)

Thank you for suffering through this! If you want to learn lots more about D3 and how to use its transitions in an actually useful way, check out my book!

d3.select("#end").each(show)
Oriole Online Tutorials
Learn alongside smart people solving hard problems

Oriole is a unique new medium that blends code, data, text, and video into a narrated learning experience with executable content.

Led by some of the most brilliant minds in technology, each lesson is an easily digestible and engaging thought-by-thought tour of the instructor’s approach to the problem in both narrative and executable code. No set-up or installation is necessary; Oriole Online Tutorials require nothing more than an internet connection and a laptop. You can write and run code within the environment. Make a mistake? Change it, and try again.

Oriole combines the expert insight and hands-on learning of in-person or online courses with the on-demand, at-your-own desk, back-up-and-run-it-again convenience of video training. You learn by doing, on your own schedule, and at your own pace.

In Oriole, we get the complete integration of video synchronized with the flow of the text, as well as the ability to execute the code: this is probably as close as we can get to learning side-by-side with Peter himself.

Fernando Perez, creator of IPython, which evolved into Project Jupyter.
**Regex golf**

Peter Norvig realizes that something is broken on the Internet. It's his job to fix it! This tutorial shows how to write Python code to play "Regex Golf."

By Peter Norvig, March 21, 2016

---

**xkcd 1313: Regex Golf**

*Peter Norvig*

*January 2014*

*revised November 2015*

I ♥ xkcd! It reliably provides top-rate insights, humor, or both. I was thrilled when I got to introduce Randall Munroe for a talk in 2007. But in xkcd #1313,
Regex Golf with Arbitrary Lists

Let's move on to arbitrary lists. I define `report`, to call `findregex`, verify the solution, and print the number of characters in the solution, the number of parts, the competitive ratio (the ratio between the lengths of a trivial solution and the actual solution), and the number of winners and losers.

```python
def report(winners, losers):
    """Find a regex to match A but not B, and vice-versa. Print summary."
    solution = findregex(winners, losers)
    verify(solution, winners, losers)
    trivial = '^' + OR(winners) + ')$'
    print('Characters: {}, Parts: {}, Competitive ratio: {:.1f},
    Winners: {}, Losers: {}' .format(
        len(solution), solution.count('|') + 1, len(trivial) /
        len(solution), len(winners), len(losers))
    return solution
report(winners, losers)
```

The top 10 boys and girls names for 2012:

```python
boys = words('jacob mason ethan noah william liam jayden michael
alexander aiden')
girls = words('sophia emma isabella olivia ava emily abigail mia
madison elizabeth')
report(boys, girls)
```
My **Dream Learning Platform** would:

- facilitate human-to-human interaction
- encourage collaboration and conversation
- provide space for play
- emphasize socialization and community development at least as much as the explicit learning
- work best for temporally bound learning communities
- handle “courses” of different lengths
- support synchronous and asynchronous communication
- serve self-motivated learners but also engage less motivated ones
- be useful for a wide range of subjects and content types
- support an array of payment mechanisms and business models
My **Dream Learning Platform** would:

- **Google Hangouts, appear.in, talky.io**
  - facilitate human-to-human interaction

- **Google Docs, Etherpad, Github, Slack**
  - encourage collaboration and conversation

- **Minecraft**
  - provide space for play

- **DIY, specialized communities**
  - emphasize socialization and community development at least as much as the explicit learning work best for temporally bound learning communities

- **Khan Academy**
  - handle “courses” of different lengths

- **Chat, audio, video, forums**
  - support synchronous and asynchronous communication

- **OSS**
  - support an array of payment mechanisms and business models
O'Reilly Media, Inc.

O'Reilly spreads the knowledge of innovators through its technology books, online services, magazines, research, and tech conferences.

Sebastopol, CA  http://www.oreilly.com

Repositories

**Learning-Path-Node-For-Frontend-Developers**

Links to example code downloads for Learning Path: Node for Frontend Developers

Updated 8 days ago

**lander** (PRIVATE)

Interactive Video & Code

Updated 9 days ago

**atlas_book_skeleton**

Skeleton files for a new Atlas project

Updated 10 days ago

**article-template**

HTML  ★ 5  ★ 6
Jupyter javascript plugin for static sites https://oreillymedia.github.io/thebe/

<table>
<thead>
<tr>
<th>zogren</th>
<th>build after merge</th>
</tr>
</thead>
<tbody>
<tr>
<td>carts</td>
<td>reorganized, added better error handling, broke stuff</td>
</tr>
<tr>
<td>examples</td>
<td>actually remove orchard url in examples</td>
</tr>
<tr>
<td>static</td>
<td>build after merge</td>
</tr>
<tr>
<td>.bowerrc</td>
<td>Add bower?</td>
</tr>
<tr>
<td>.gitignore</td>
<td>Remove components from GH</td>
</tr>
<tr>
<td>LICENSE</td>
<td>added mit license</td>
</tr>
<tr>
<td>README.md</td>
<td>removed links to examples that won't work on tmpnb.org, documented that</td>
</tr>
<tr>
<td>bower.json</td>
<td>Add bower?</td>
</tr>
<tr>
<td>index.html</td>
<td>typo</td>
</tr>
</tbody>
</table>

Thebe takes the Jupyter (formerly ipython) front end, and make it work outside of the notebook context.

What? Why?

Let’s discuss

bit.ly/28Aec60

My Dream Learning Platform would:

> facilitate human-to-human interaction
> encourage collaboration and conversation
> provide space for play
> emphasize socialization and community development at least as much as the explicit learning work best for temporally bound learning communities
> handle “courses” of different lengths
> support synchronous and asynchronous communication
> serve self-motivated learners but also engage less motivated ones
> be useful for a wide range of subjects and content types
> support an array of payment mechanisms and business models