In computer science, data can be represented in a lot of different ways, and naturally, every single one of them has its advantages as well as disadvantages in certain fields.

Since computers are unable to process categorical data as these categories have no meaning for them, this information has to be prepared if we want a computer to be able to process it.

This action is called preprocessing. A big part of preprocessing is encoding - representing every single piece of data in a way that a computer can understand (the name literally means "convert to computer code").

In many branches of computer science, especially machine learning and digital circuit design, One-Hot Encoding is widely used.

In this article, we will explain what one-hot encoding is and implement it in Python using a few popular choices, Pandas and Scikit-Learn. We'll also compare its effectiveness to other types of representation in computers, its strong points and weaknesses, as well as its applications.
1. You want your publications to look good, we now make it easy to get your graphs in exactly the size you need.

- **The Weekly Challenge #054** [4]
  
  For the first time, since I started participating the weekly challenges, I thought of doing one-liner. With handy CPAN modules, it was pretty straight forward in Perl. Even Raku with built-in features wasn’t far behind Perl. Like in the past, I learn something new in Raku every week. This week was no different. I will share what I learnt this time later.

- **How to compare objects in PHP** [5]
  
  PHP offers a simple way to compare objects using the comparison (==) and identity (===) operators.
  
  When using the comparison operator (==), object variables are compared in a simple manner: Two object instances are equal if they have the same attributes and values and are instances of the same class.

- **Fix Class ?DOMDocument? not found error** [6]

- **How JAMstack Is Shaking Up Static Application Development** [7]
  
  In an API-driven world that is increasingly mobile, JAMstack is well-positioned to become a de facto method for application architecture and delivery.

**Development**

**Source URL:** [http://www.tuxmachines.org/node/135896](http://www.tuxmachines.org/node/135896)

**Links:**