最直观的感受：对象存储是一种将要存储的东西当作一个对象(Object)存储在磁盘上，并不考虑它的类型、大小，用一个唯一的标识符表示该对象，以便再次读取、更新或删除该对象的存储系统。

用户不能直接在磁盘上访问对象，对象存储面向具体的应用，而非用户，应用利用对象存储系统提供的API访问对象。

与文件系统相比，对象存储需要存储的元数据更大，但是由于其偏平目录结构设计，也使得对象存储更易扩容与重新均衡。

对象存储的特征：

远程访问、海量用户、无限扩容、廉价（不需要采用定制硬件）。

对象存储是支持海量用户远程访问的无限容量廉价存储系统。

**对象存储**是无层次结构的数据存储方法，通常用于云中。基于对象的存储不使用目录树。各个单独的数据（对象）单元存在于存储池中的同一级别。每个对象都有唯一的识别名称，供应用进行检索。此外，每个对象可包含有助于检索的元数据。

对象存储注意事项

* **功能有限**

用于访问数据的对象 API 通常非常简单。应用必须实现更复杂的数据管理需求。

* 兼容性

如果不添加额外的层次，文件系统工具（例如，POSIX 实用程序）就无法与基于对象的存储系统进行互动。

* **数据类型**

对象存储是非结构化数据（例如，媒体和 Web 内容）的理想之选。它不适合需要定期修改的数据。

* **API 支持**

应用必须进行更新才能充分利用对象存储的优势。许多供应商都在发布含有内置支持的修订版。

虽然HDFS与Openstack对象存储（Swift）之间有着一些相似之处，但是这两种系统的总体设计却大不一样。

　　1. HDFS使用了**中央系统**来维护文件元数据（Namenode，名称节点），而在Swift中，元数据呈分布式，跨集群复制。使用一种中央元数据系统对HDFS来说无异于**单一故障点**，因而扩展到规模非常大的环境显得更困难。

　　2. Swift在设计时考虑到了**多租户架构**，而HDFS没有多租户架构这个概念。（提高并发量）

　　3. HDFS针对**更庞大的文件**作了优化（这是处理数据时通常会出现的情况），Swift被设计成了可以**存储任何大小的文件**。

　　4. 在HDFS中，文件写入一次，而且每次只能有一个文件写入；而在Swift中，文件可以写入多次；在并发操作环境下，以最近一次操作为准。

　　5. HDFS用Java来编写，而Swift用Python来编写。

另外，HDFS被设计成了可以存储数量中等的大文件，以支持数据处理，而Swift被设计成了一种比较通用的存储解决方案，能够可靠地存储数量非常多的大小不一的文件。