

基于NumPy构建，利用它的高级数据结构和操作工具，可使数据分析工作变得更加便捷高效。

符号标记

s | 一个Series对象.

df | 一个DataFrame对象.

导入包 (Pandas 0.20.1)

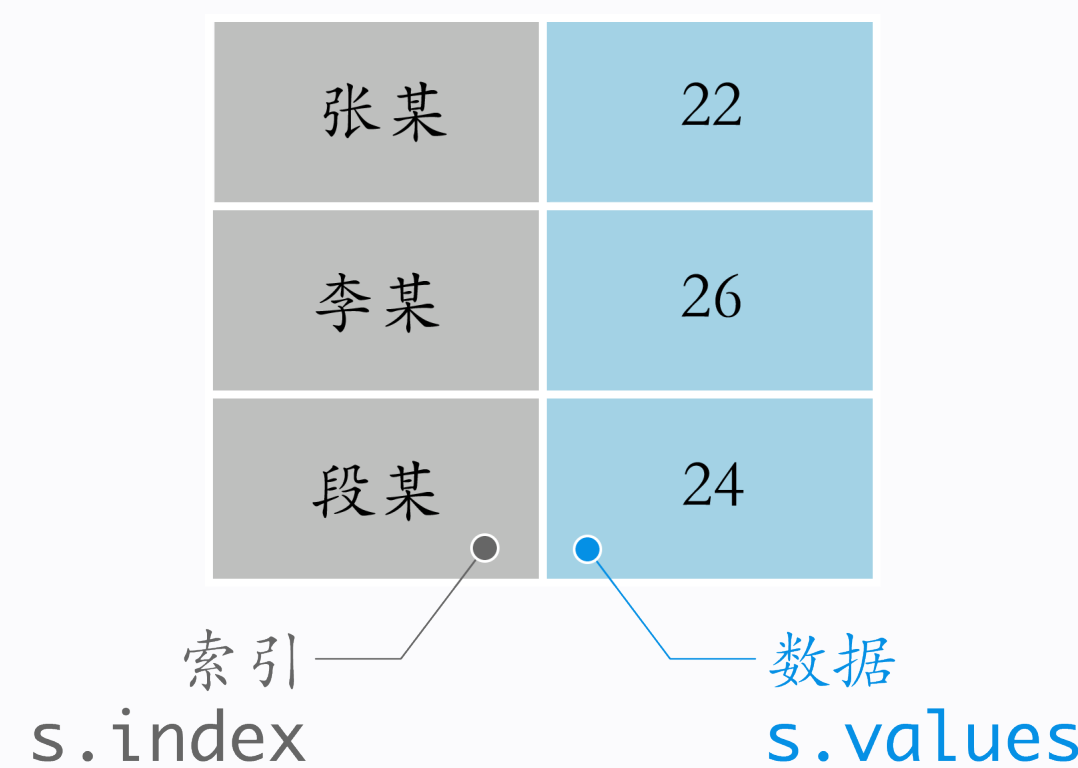
import numpy as np

import pandas as pd

## 基本数据结构

### Series

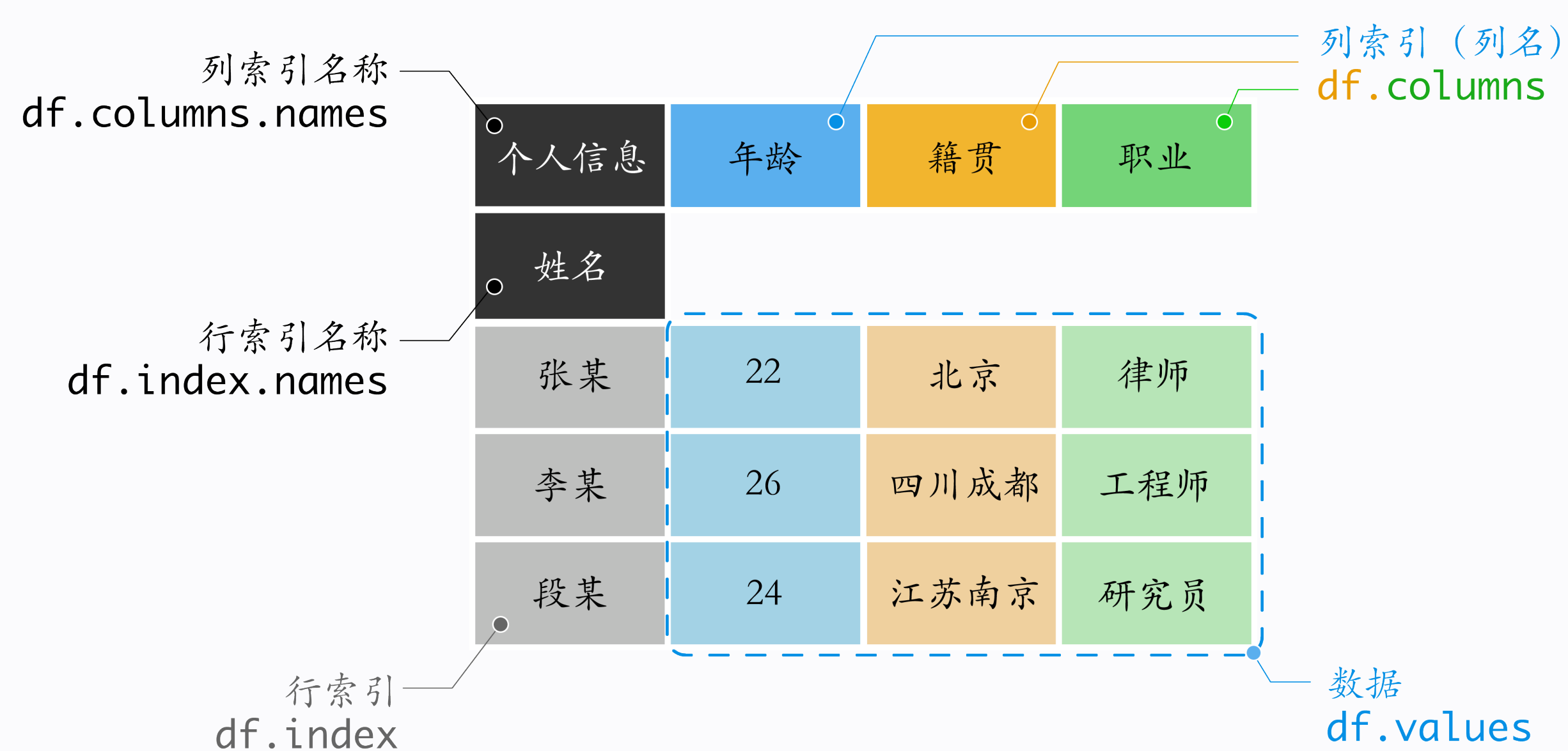
一维数据结构，包含行索引和数据两个部分。



```
s = pd.Series([14, 15, 17],
              index=[u'张某', u'李某', u'段某'])
```

### DataFrame

二维数据结构，包含带索引的多列数据，各列的数据类型可能不同。



```
df = pd.DataFrame([[22, u'北京', u'律师' ],
                   [26, u'四川成都', u'工程师'],
                   [24, u'江苏南京', u'研究员']],
                  index=[u'张某', u'李某', u'段某'],
                  columns=[u'年龄', u'籍贯', u'职业'])
```

## 文件读写

从文件中读取数据 (DataFrame)

`pd.read_csv()` | 从CSV文件读取.

参数: **file**, 文件路径; **sep**, 分隔符; **index\_col**, 用作行索引的一列或者多列; **usecols**, 选择列; **encoding**, 字符编码类型, 通常为 'utf-8'.

`pd.read_table()` | 从制表符分隔文件读取, 如TSV.

`pd.read_excel()` | 从Excel文件读取.

`pd.read_sql()` | 从SQL表或数据库读取.

`pd.read_json()` | 从JSON格式的URL或文件读取.

`pd.read_clipboard()` | 从剪切板读取.

将DataFrame写入文件

`df.to_csv()` | 写入CSV文件.

参数: **file**, 文件路径; **sep**, 分隔符; **columns**, 写入文件的列; **header**, 是否写入表头; **index**, 是否写入索引.

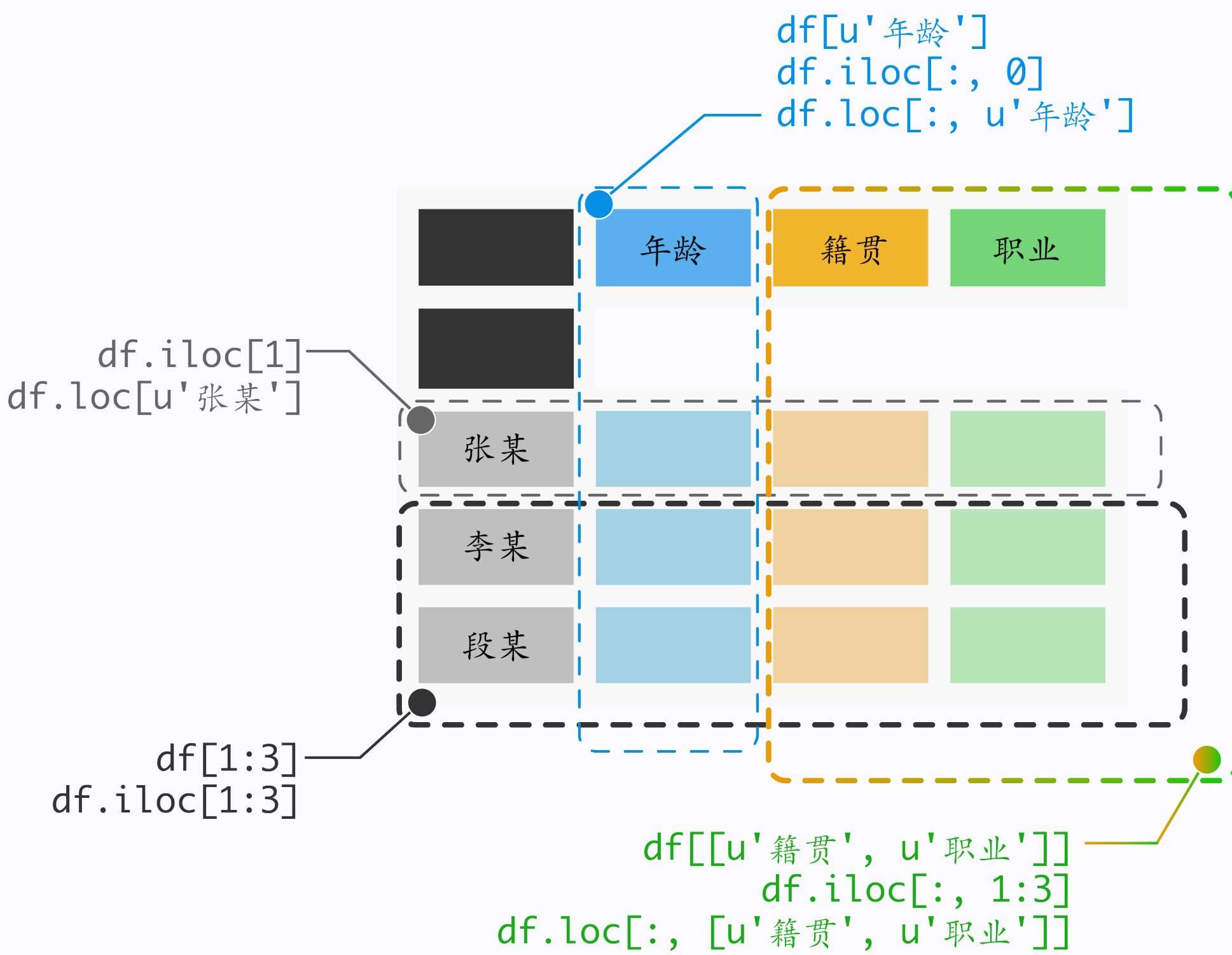
`df.to_excel()` | 写入Excel文件.

`df.to_sql()` | 写入SQL表或数据库.

`df.to_json()` | 写入JSON格式的文件.

`df.to_clipboard()` | 写入剪切板.

## 数据索引



`df[5:10]` | 通过切片方式选取多行.

`df[col_label]` or `df.col_label` | 选取列.

`df.loc[row_label, col_label]` | 通过标签选取行/列.

`df.iloc[row_loc, col_loc]` | 通过位置 (自然数) 选取行/列.

## 数据探索

`s.unique()` | 唯一值.

`s.value_counts()` | 唯一值及其计数.

`df.head(n)` | 前n行数据.

`df.tail(n)` | 后n行数据.

`df.sample(n)` | 随机采样的n行数据.

`df.shape` | 行数和列数.

`df.info()` | 样本数、数据类型和内存占用等信息.

`df.describe()` | 描述性统计信息汇总.

## 筛选与过滤

`df[u'职业'].isin([u'研究员'])` | 判断“职业”是否包含“研究员”.

`df[df[u'年龄'] > 25]` | 逻辑表达式筛选行.

`df.loc[df[u'年龄'] > 25, [u'职业']]` | 逻辑表达式筛选行, 并筛选列.

`df.filter('regex')` | 筛选满足表达式的列.

`df.drop([u'李某'], axis=0)` | 删除“李某”行.

`df.drop([u'年龄'], axis=1)` | 删除“年龄”列.

`df.drop_duplicates()` | 删除重复样本.

`df.select_dtypes()` | 根据数据类型包含(include)或去除(exclude)列.

## 调整索引和修改标签

`inplace=True`, 更改原始Pandas对象;

`axis=0`, 纵向轴; `axis=1`, 横向轴.

`df.astype({u'年龄': 'int8'})` | 数据类型转换.

`df.index = [a, b, c]` | 修改行索引.

`df.columns = [a, b, c]` | 修改列索引.

`df.set_index()` | 设置某一列为索引.

`df.reset_index()` | 重置索引为自然数索引.

`df.reindex(index, columns=[A, B])` | 根据index和columns进行重排.

`df.rename(index, columns={A: B})` | 对行索引或列标签进行修改.

## 排序和排名

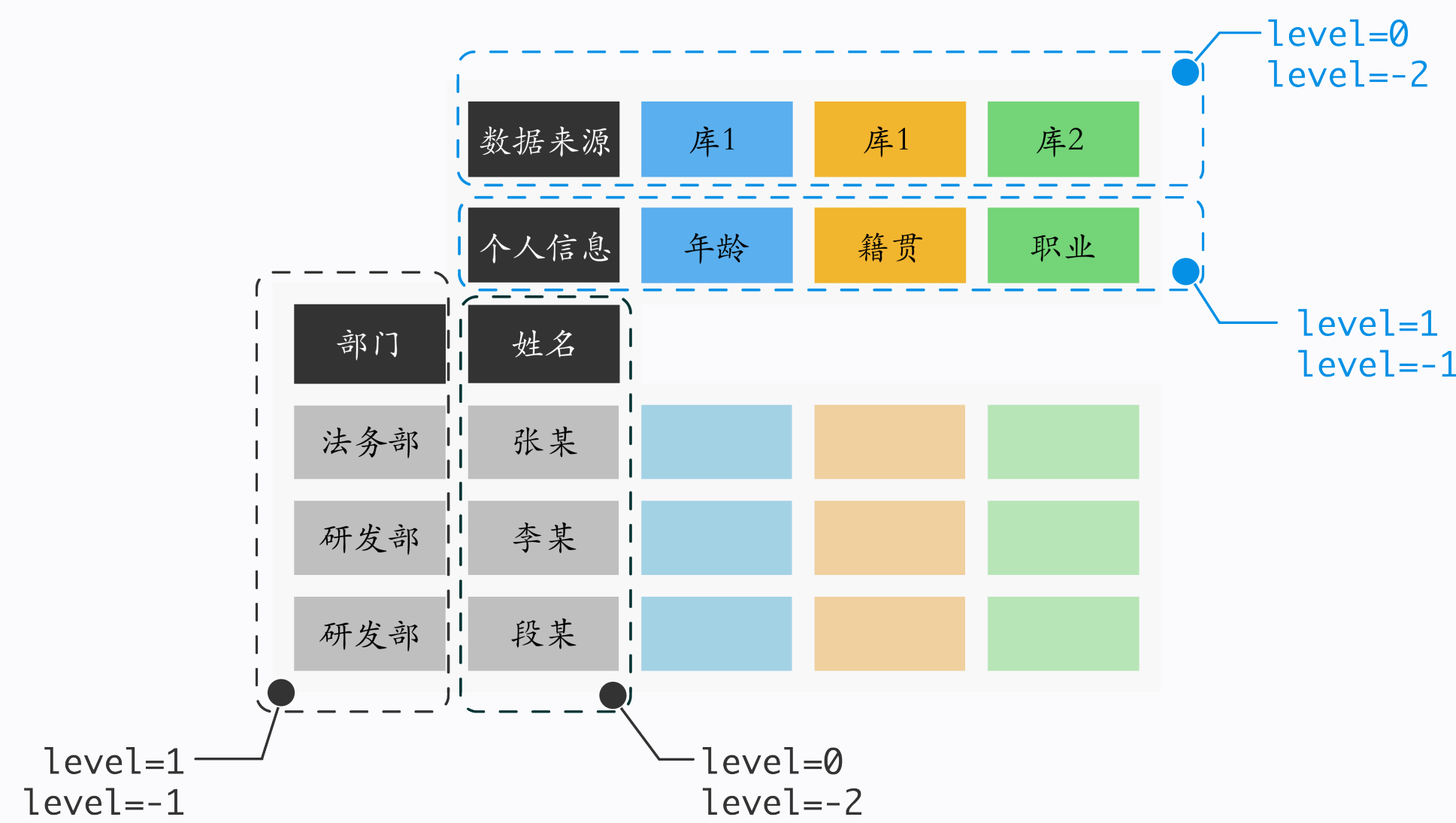
`df.sort_index(axis=0)` | 按照行索引进行排序.

`df.sort_values(A, axis=0)` | 根据“A”列值进行升序排列.

`df.sort_values(A, ascending=False)` | 根据“A”列值进行降序排列.

`df.rank(axis=0)` | 返回元素在所属列的排名.

## 层次化索引



`df.index = [[u'法务部', u'研发部', u'研发部'], [u'张某', u'李某', u'段某']]` | 创建层次化索引.

`df.index.names = [u'部门', u'姓名']` | 创建行索引名称.

`df.swaplevel(0, 1)` | 互换行索引级别.

`df.reorder_levels([1, 0])` | 重排行索引级别.

`df.xs(u'年龄', level=1, axis=1)` | 根据某一层级进行交叉选择.

## 统计函数

`df.sum()` | 求和.

`df.mean()` | 均值.

`df.max()` | 最大值.

`df.min()` | 最小值.

`df.median()` | 中位数.

`df.quantile([0.25, 0.75])` | 分位数.

`df.std()` | 标准差.

`df.var()` | 方差.

`df.cumsum()` | 累和.

`df.mode()` | 众数.

`df.corr()` | 相关系数矩阵.

`df.cov()` | 协方差矩阵.

`df.corrwith()` | 不同Pandas对象之间的相关性.

## 缺失值检测与处理

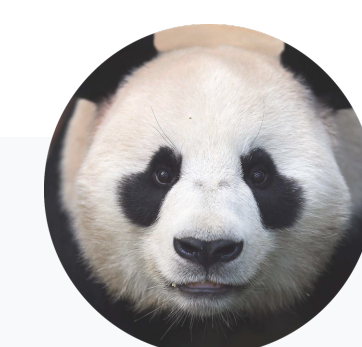
`df.isnull()` | 判断每个元素是否为缺失值.

`df.notnull()` | 判断每个元素是否为非缺失值.

`df.fillna({A: 1, B: 22})` | 多列缺失值填补.

`df.dropna()` | 删除缺失值所在行/列.

参数: **axis**, 操作轴, 默认为横向, 即丢弃行; **how**, 丢弃方式; **subset**, 考虑部分行和列; **thresh**, 非缺失值数量下限.





## 数据合并

### 数据融合（merge）

	姓名	年龄
0	张某	22
1	李某	26
2	段某	24

	姓名	籍贯
7	张某	北京
8	李某	四川成都
9	钱某	江苏南京

	姓名	年龄	籍贯
0	张某	22	北京
1	李某	26	四川成都

`pd.merge(left, right, how='inner', on=u'姓名')`

	姓名	年龄	籍贯
0	张某	22.0	北京
1	李某	26.0	四川成都
2	段某	24.0	NaN
3	钱某	NaN	江苏南京

`pd.merge(left, right, how='outer', on=u'姓名')`

	姓名	年龄	籍贯
0	张某	22	北京
1	李某	26	四川成都
2	段某	24	NaN

`pd.merge(left, right, how='left', on=u'姓名')`

	姓名	年龄	籍贯
0	张某	22.0	北京
1	李某	26.0	四川成都
2	钱某	NaN	江苏南京

`pd.merge(left, right, how='right', on=u'姓名')`

`pd.merge(left, right)` | 类数据库的数据融合操作.

参数: **how**, 融合方式, 包括左连接、右连接、内连接（默认）和外连接; **on**, 连接键; **left\_on**, 左键; **right\_on**, 右键; **left\_index**, 是否将left行索引作为左键; **right\_index**, 是否将right行索引作为右键.

### 数据融合（join）

姓名	年龄
张某	22
李某	26
段某	24

姓名	籍贯
张某	北京
李某	四川成都
钱某	江苏南京

姓名	年龄	籍贯
张某	22	北京
李某	26	四川成都
段某	24	NaN

等价于 `pd.merge(left, right, left.join(right), how='left', left_index=True, right_index=True)`

`left.join(right)` | 在索引上的数据融合操作.

### 数据融合（combine\_first）

`left.combine_first(right)`

在数据融合的同时（行索引和列索引取并集），使用right中的值填补left中相应位置的缺失值.

### 轴向连接

	df1		

	df2	



`pd.concat([df1, df2], axis=0)`



`pd.concat([df1, df2], axis=1)`

`pd.concat([df1, df2])` | 轴向连接多个DataFrame.

## 轴向旋转和数据转换

### 轴向旋转

`df.stack()` | 将数据的列“旋转”为行.

`df.unstack()` | 将数据的行“旋转”为列.

### 数据转换

`s.map()` | 利用函数或映射进行数据转换.

`df.applymap()` | 利用函数或映射进行数据转换.

`df.replace()` | 替换元素值.

`df.columns.map(str.upper)` | 将列名转换为大写.

## 数据聚合与分组运算

`df.groupby('A')` | 根据“A”列的值进行分组，并返回一个GroupBy对象.

`df.groupby(['A', 'B'])` | 根据“A”列和“B”列的值组合进行分组，并返回一个GroupBy对象.

`df.groupby('A')['B'].mean()` | 根据“A”列的值进行分组，并计算每一组“B”的均值.

`df.groupby('A').agg(np.mean)` | 根据“A”列的值进行分组，并计算每一组中各列的均值.

`df.apply(np.mean)` | 对DataFrame的每一列求均值.

`df.apply(np.mean, axis=1)` | 对DataFrame的每一行求均值.

`df.pivot_table(index='A', values=['B', 'C'], aggfunc=mean)` | 创建数据透视表，根据“A”列的值进行分组，并计算每一组中“B”和“C”的均值.

## 离散化和哑变量编码

`pd.cut(s, bins=[0, 5, 10], labels=['A', 'B'])` | 离散化和面元划分.

`pd.qcut(s, 2, labels=['A', 'B'])` | 等频离散化.

`pd.get_dummies(df, columns=['A'], drop_first=True)`

对“A”列进行哑变量编码，并去掉第一个编码特征.

## 串联方法

通过调用多个Pandas方法将多个处理过程串联起来.

`pd.merge(left, right, how='outer', indicator=True).query('_merge == "left_only") .drop(['_merge'], axis=1)`  
保留在left中但不在right中的行.

## 文本数据规整

`s.str.method`

矢量化字符串方法，可跳过缺失值，遇缺失值将不报错.

`s.str.lower()` | 将所有元素转换为全部小写.

`s.str.upper()` | 将所有元素转换为全部大写.

`s.str.replace()` | 替换值.

`s.str.split(" ").str[0]` | 空格分割，并取列表中的第一个元素.

`s.str.split(" ").str.get(0)` | 空格分割，并取列表中的第一个元素.

`s.str.split(" ", expand=True)` | 空格分割，并扩展为多列.

`s.str.contains(r'\d')` | 利用正则表达式判断是否元素包含数字.

`s.str.extract(r'(\d.)', expand=False)` | 利用正则表达式提取每个元素中的数字.

## 数据可视化

`df.plot(kind='hist')` | 设置kind参数绘制直方图.

`df.plot.line()` | 调用对象接口绘制折线图.

`df.plot.scatter('x', 'y')` | 绘制x-y散点图.

包括图形: 面积图`df.plot.area`; 纵向柱状图`df.plot.bar`; 横向柱状图`df.plot.barh`; 盒图`df.plot.box`; 和核密度图`df.plot.kde`; 蜂巢图`df.plot.hexbin`; 饼图`df.plot.pie`.

