

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1. jydata-common

1.1. 一、jy_code

1.2. 二、jy_conf 公共配置

1.3. 三、jy_crypt 加解密

1.3.1. 1. base_bit_num 基础数位

1.3.1.1. a. 类定义

```
class BaseBitNum(object):  
  
    def __init__(self, num_bit=10):  
        """  
        初始化  
        :param num_bit: 数位 2/8/10/16/32...  
        """  
  
    @property  
    def num_bit(self):  
        """ 属性方法: 获取数位 """  
  
    @property  
    def bit_chars(self):  
        """ 属性方法: 字符序列 """
```

1.3.1.2. b. 已定义对象

```
from jy_crypt.base_bit_num import BaseBitNum  
BINARY = BaseBitNum(num_bit=2) # 二进制  
OCTAL = BaseBitNum(num_bit=8) # 8进制  
NUM = BaseBitNum(num_bit=10) # 10进制  
HEX = BaseBitNum(num_bit=16) # 16进制  
BASE32 = BaseBitNum(num_bit=32) # 32进制  
BASE64 = BaseBitNum(num_bit=64) # 64进制
```

1.3.2. 2. bit_num 进制数

1.3.2.1. a. 类定义

```
from jy_crypt.base_bit_num import BaseBitNum, NUM  
class BitNum(object):  
    def __init__(self, val, bit=NUM):  
        """  
        初始化  
        :param val: 值  
        :param bit: 数位  
        """  
  
    def convert_to_bit(self, aim_bit: BaseBitNum):  
        """  
        进制转换  
        :param aim_bit: 目标数位  
        :return:  
        """  
  
    @property  
    def val_num(self) -> int:  
        """  
        对应的十进制数值
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle/MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.3.1. a. 条件类
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
:return:
"""

def get_bit_char(self, m):
    """
    数值转字符
    :param m: 数值
    :return:
    """

def get_bit_val(self, ch):
    """
    字符转数值
    :param ch: 字符
    :return:
    """
```

例

```
from jy_crypt.base_bit_num import BASE32
from jy_crypt.bit_num import BitNum
_32_bit_num = BitNum(245, BASE32)
```

3.3. 3. 佳缘线上加解密

3.3.1. a. 身份证号加密

```
def encrypt_idc(idc: str):
    """
    加密线上身份证号
    :type idc: str 待加密的身份证号
    """
```

```
from jy_crypt.jyol_crypt import encrypt_idc
encrypt_idc('350122198910116847')
### 'fhcdeedlkldzddikgj'
```

1.3.3.2. b. 身份证号解密

```
def decrypt_idc(en_idc: str):
    """
    解密线上身份证号
    :param en_idc: 待解密的身份证号密文
    :return:
    """
```

示例

```
from jy_crypt.jyol_crypt import decrypt_idc
decrypt_idc('fhcdeedlkldzddikgj')
### '350122198910116847'
```

1.3.3.3. c. 姓名加解密

```
def crypt_name(name: str):
    """
    加解密姓名
    :param name: 待加(解)密的姓名(密文)
    :return:
    """
```

示例

```
from jy_crypt.jyol_crypt import crypt_name
crypt_name('诸葛亮')
### '嚮蓂仪'
crypt_name('嚮蓂仪')
### '诸葛亮'
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL/Hive/Impala
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL/Hive/Impala
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle/MySQL/Hive/Impala
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1.3.4. 4. 佳缘vip加解密

1.3.4.1. a. 佳缘vip数据加解密

```
def convert(in_str: str, convert_type: int, dept=None):
    """
    加解密（数位转换）
    :param in_str: 待加（解）密字符串
    :param convert_type: 1:解密, 0:加密
    :param dept:
    :return:
    """
```

示例

```
from jy_crypt.jyvip_convert import convert
## 解密
:convert('1j1m1i1k1j1g1h1p1p1j1g1h1h1g1g1g1h1p', 1)
## '362430199301100019'
## 加密
:convert('362430199301100019', 0)
## '1j1m1i1k1j1g1h1p1p1j1g1h1h1g1g1g1h1p'
```

4. 四、jy_db 数据库

1.4.1. 1. 获取数据库对象

4.1.1. a. 查看数据库配置信息

```
from jy_db import show_db_confs
show_db_confs(db_type='Orcl') # db_type: Oracle/MySQL/Hive/Impala
```

4.1.2. b. 获取数据库对象

```
from jy_db import get_db
orcl_dc = get_db(db_type='Orcl', key='dc') # db_type同上, key为数据库配置信息中的键
orcl_huawei = get_db(db_type='Orcl', key='huawei')
mysql_crm = get_db(db_type='MySQL', key='crm')
hive_cloud = get_db(db_type='Hive', key='cloud')
impala_cloud = get_db(db_type='Impala', key='cloud')
```

1.4.2. 2. 数据库对象api

1.4.2.1. a. select: 查询, 适用于Oracle/MySQL/Hive/Impala

1.4.2.1.1. 方法定义

```
def select(self, sql, cursor=None, size=-1, params: tuple = None, key_pre='#', **kw):
    """
    查询数据
    :type size: int 抓取结果条数, 默认取全部查询结果
    :param sql: select sql
    :param cursor:
    :param params: params of select sql
    :param key_pre: 命名参数标识符, 默认为#
    :param kw: 命名参数
    :return:
    """
```

1.4.2.1.2. 示例

```
from jy_db import get_db
orcl_dc = get_db('Orcl', 'dc')
mysql_crm = get_db('MySQL', 'crm')
# 完整sql, 无参数
data = orcl_dc.select("select * from city where city_new = '北京'")
"""
```

Table of Content

1. jydata-common

1.1. 一、jy_code

1.2. 二、jy_conf 公共配置

1.3. 三、jy_crypt 加解密

1.3.1. 1. base_bit_num 基础数位

1.3.1.1. a. 类定义

1.3.1.2. b. 已定义对象

1.3.2. 2. bit_num 进制数

1.3.2.1. a. 类定义

1.3.3. 3. 佳缘线上加解密

1.3.3.1. a. 身份证号加密

1.3.3.2. b. 身份证号解密

1.3.3.3. c. 姓名加解密

1.3.4. 4. 佳缘vip加解密

1.3.4.1. a. 佳缘vip数据加解密

1.4. 四、jy_db 数据库

1.4.1. 1. 获取数据库对象

1.4.1.1. a. 查看数据库配置信息

1.4.1.2. b. 获取数据库对象

1.4.2. 2. 数据库对象api

1.4.2.1. a. select: 查询, 适用于Oracle/

1.4.2.1.1. 方法定义

1.4.2.1.2. 示例

1.4.2.2. b. update: 增删改, 适用于Orac

1.4.2.2.1. 方法定义

1.4.2.2.2. 示例

1.4.2.3. c. update_batch: 批量更新, 一

1.4.2.3.1. 方法定义

1.4.2.3.2. 示例

1.5. 五、jy_entity 实体基类

1.6. 六、jy_factory 工厂类

1.7. 七、jy_result 返回结果

1.8. 八、jy_util 工具/方法

1.8.1. 1. attr_utils: 对象属性工具方法库

1.8.1.1. a. 设置对象属性

1.8.2. 2. calender_utils: 日历工具方法库

1.8.2.1. a. 获取给定日期所在月份的工作日

1.8.3. 3. condition_utils: 条件类工具方法库

1.8.4. 4. date_utils: 日期时间类工具方法库

1.8.4.1. a. 常量定义

1.8.4.2. b. 格式化时间元素数字

1.8.4.3. c. 时间戳转datetime

注意事项:

1> 查询sql结尾不添加分号 (;), 下同。

"""

f params -- 参数列表, 必须是tuple类型: (param1, param2, ...)

```
data = orcl_dc.select("select * from city where city_new = '%s' and id < %s", params=('北京', 111))
```

"""

主意事项:

> 查询sql所有参数占位符必须是%s(int/float类型的参数也不例外, 不采用%d/%f形式占位符), 如果查询参数类型是

> 如果参数较多(>4)时, 建议使用关键字参数形式。

> 如果查询sql中有'%', 不建议使用参数列表(params), 可改用关键字参数形式

> 特殊参数(系统定义, 在执行时会被转化成特定的值, 同样适用于命名参数形式), 要传确定的参数时应避开这些特

特殊参数格式: 类型(非必须)#可执行方法名?k=v(可执行方法参数, 是否必须要看可执行方法名对应的方法是否有)

I: 日期时间字符串(datestr#{func_name}?k=v)

指定日期(默认今天)开始时间: datestr#day_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)结束时间: datestr#day_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的昨天开始时间: datestr#yesterday_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的昨天结束时间: datestr#yesterday_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的月初开始时间: datestr#month_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的月末结束时间: datestr#month_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的年初开始时间: datestr#year_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的年末结束时间: datestr#year_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

II: 日期时间戳(datets#{func_name}?k=v)

指定日期(默认今天)开始时间: datets#day_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)结束时间: datets#day_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的昨天开始时间: datets#yesterday_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的昨天结束时间: datets#yesterday_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的月初开始时间: datets#month_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的月末结束时间: datets#month_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的年初开始时间: datets#year_begin[?day={datetime}|year=%dmonth&month=%d&days=%d]

指定日期(默认今天)的年末结束时间: datets#year_end[?day={datetime}|year=%dmonth&month=%d&days=%d]

III: 无类型({func_name}?k=v)

暂无, 待添加

"""

kw -- 命名参数

```
data = orcl_dc.select("select * from city where city_new = '#name' and id < #id", name='北京', id=111)
```

"""

主意事项:

> 查询sql的查询参数类型是varchar/string时, 对应的占位参数名应包含在引号内。

2> 命名参数可以和params同时使用, eg:

```
data = orcl_dc.select("select * from city where city_new = '%s' and id < #id", params=('北京',), id=111)
```

3> 如果默认命名参数标识符(#)和查询参数冲突时可以替换命名参数标识符(传入新的key_pre), eg:

```
data = orcl_dc.select("select * from city where city_new = '?name' and id < ?id", key_pre='?', name='北京', id=111)
```

"""

size -- 抓取结果条数

```
data = mysql_crm.select('select con_id, contract_num from contract order by con_id desc', size=5)
```

"""

主意事项:

1> 该参数不适用于Oracle。

2> 该参数可以和以上参数同时使用。

"""

1.4.2.2. b. update: 增删改, 适用于Oracle/Mysql/Hive/Impala(前提是有相应的数据库权限)

1.4.2.2.1. 方法定义

```
def update(self, sql, params=None, cursor=None, **kw):
    """
    insert/update/delete data
    :param sql: insert/update/delete sql
    :param params: params of update sql
    :param cursor:
    :return:
    """
```

1.4.2.2.2. 示例

```
# update
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle/MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1.4.2.3. c. update_batch: 批量更新, 一般用于批量Insert

1.4.2.3.1. 方法定义

```
def update_batch(self, sql, params=None, batch_num=10000, cursor=None):
    """
    batch insert/update/delete data
    :param sql: insert/update/delete sql
    :param params: batch params(list)
    :param batch_num: batch num, the num of per commit, default 10000
    :param cursor:
    :return:
    """
```

1.4.2.3.2. 示例

5. 五、jy_entity 实体基类

6. 六、jy_factory 工厂类

7. 七、jy_result 返回结果

8. 八、jy_util 工具/方法

1.8.1. 1. attr_utils: 对象属性工具方法库

1.8.1.1. a. 设置对象属性

方法定义

```
def set_attrs(o, **kwargs):
    """
    设置对象属性, 用于将dict赋值给对象属性
    :param o: 待设置属性对象
    :param kwargs: 需要设置的属性键值对
    :return: None
    """
```

示例

```
from jy_utils.attr_utils import set_attrs
class A:
    attr_1 = 0
    attr_2 = 0

a = A()
a.attr_1, a.attr_2
### (0, 0)
set_attrs(a, **{'attr_1': 1, 'attr_2': 2})
a.attr_1, a.attr_2
### (1, 2)
```

1.8.2. 2. calender_utils: 日历工具方法库

1.8.2.1. a. 获取给定日期所在月份的工作日序号list和休息日序号list

方法定义

```
from datetime import datetime
def month_work_days(day: datetime, weekends=(5, 6), holidays: list = None, workdays: list = None)
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
"""
获取给定日期所在月份的工作日序号list和休息日序号list
:param day: 给定日期
:param weekends: 固定周休序号, 默认为(5, 6), 对应为(周六,周日)
:param holidays: 自定义节假日(list)
:param workdays: 自定义工作日(list), 调班
:return: (工作日序号list,休息日序号list)
"""
```

例

```
from datetime import datetime
from jy_utils.calender_utils import month_work_days, month_nrd_work_day

# 无自定义节假日和自定义调休日
## 默认周休(周六、日双休)
result = month_work_days(datetime.now())
result: (
    [2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, 30, 31],
    [1, 7, 8, 14, 15, 21, 22, 28, 29]
)

## 非默认周休(例如周一、二双休)
result = month_work_days(datetime.now(), weekends=(0,1))
result: (
    [1, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29],
    [2, 3, 9, 10, 16, 17, 23, 24, 30, 31]
)

# 自定义节假日和调休日
result = month_work_days(datetime(2021, 10, 1), workdays=(9,), holidays=(1,2,3,4,5,6,7))
result: (
    [8, 9, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29],
    [1, 2, 3, 4, 5, 6, 7, 10, 16, 17, 23, 24, 30, 31]
)
```

1.8.3. 3. condition_utils: 条件类工具方法库

1.8.4. 4. date_utils: 日期时间类工具方法库

1.8.4.1. a. 常量定义

```
from datetime import datetime

# 单位值
MONTHS = 12
HOURS = 23
MINUTES = 59
SECONDS = 59

# 日期时间格式
DATE_FMT = '%Y-%m-%d'
DATE_FMT0 = '%y-%m-%d'
DATE_FMT_HMS = '%Y-%m-%d %H:%M:%S'
DATE_FMT_IMS = '%Y-%m-%d %I:%M:%S'
DATE_FMT0_HMS = '%y-%m-%d %H:%M:%S'
DATE_FMT0_IMS = '%y-%m-%d %I:%M:%S'
DATE_FMT_Ymd = '%Y%m%d'
DATE_FMT_ymd = '%y%m%d'
DATE_FMT_Ymd_HMS = '%Y%m%d %H:%M:%S'
DATE_FMT_Ymd_IMS = '%Y%m%d %I:%M:%S'
DATE_FMT_ymd_HMS = '%y%m%d %H:%M:%S'
DATE_FMT_ymd_IMS = '%y%m%d %I:%M:%S'
DAY_BEGIN = '00:00:00'
DAY_BEGIN12 = '00:00:00 am'
DAY_END = '23:59:59'
DAY_END12 = '23:59:59'

# 特殊时间点定义
ZERO_DATETIME = datetime(1970, 1, 1)
DATE_20190101 = datetime(2019, 1, 1)
DATE_20210201 = datetime(2021, 2, 2)
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1.8.4.2. b. 格式化时间元素数字

方法定义

```
def format_numbers(*numbers: tuple, min_len=2, max_len=0, fill_num=0, fill_loc=1, sub_loc=-1):
    """
    格式化时间元素数字
    :param numbers: 带格式化数字组
    :param min_len: 最小长度
    :param max_len: 最大长度
    :param fill_num: 不足最小长度时填充的数字
    :param fill_loc: 填充位置: 1: 左填充; -1: 右填充
    :param sub_loc: 超出长度截取位置: 1: 左截取; -1: 右截取
    :return:
    """
```

例

```
from jy_utils.date_utils import format_numbers
format_numbers(2021, 9, 23, 1, 29, 6)
## ('2021', '09', '23', '01', '29', '06')
format_numbers(2021, 9, 23, 1, 29, 6, max_len=2)
## ('21', '09', '23', '01', '29', '06')
format_numbers(2021, 9, 23, 1, 29, 6, max_len=2, sub_loc=1)
## ('20', '09', '23', '01', '29', '06')
```

1.8.4.3. c. 时间戳转datetime

方法定义

```
def from_timestamp(timestamp):
    """
    时间戳转datetime
    :param timestamp: 时间戳
    :return:
    """
```

例

```
from jy_utils.date_utils import from_timestamp, now_timestamp
from_timestamp(0)
## datetime.datetime(1970, 1, 1, 8, 0)
now_timestamp()
## 1632374848.482117
from_timestamp(1632374848.482117)
## datetime.datetime(2021, 9, 23, 13, 27, 28, 482117)
```

1.8.4.4. d. 当前时间

方法定义

```
def now():
    """获取当前时间"""
```

示例

```
from jy_utils.date_utils import now
now()
## datetime.datetime(2021, 9, 23, 13, 35, 56, 752159)
```

1.8.4.5. e. 当前时间戳

方法定义

```
def now_timestamp():
    """获取当前时间戳"""
```

示例

```
from jy_utils.date_utils import now_timestamp
now_timestamp()
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询，适用于Oracle
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改，适用于Oracle
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新，适用于Oracle
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1632375409.143204

1.8.4.6. f. 当前时间字符串

法定义

```
def now_str(fmt):
    """
    获取当前时间字符串
    :param fmt: 默认DATE_FMT_HMS
    :return:
    """
```

例

```
from jy_utils.date_utils import now_str
now_str()
### '2021-09-23 13:39:10'
now_str(fmt='%Y%m%d')
### '20210923'
```

1.8.4.7. g. 当前日期字符串

法定义

```
def cur_day_str(fmt):
    """
    获取当前日期字符串
    :param fmt: 默认DATE_FMT
    :return:
    """
```

```
from jy_utils.date_utils import cur_day_str
cur_day_str()
### '2021-09-23'
cur_day_str(fmt='%Y%m%d')
### '20210923'
```

1.8.4.8. h. 获取给定时间的的时间戳

方法定义

```
def day_ts(day=None):
    """
    获取给定时间的的时间戳
    :param day: 默认为当前时间
    :return:
    """
```

示例

```
from jy_utils.date_utils import day_ts, datetime
day_ts()
### 1632375914.375838
day_ts(day=datetime(2021, 1, 1))
### 1609430400.0
```

1.8.4.9. i. 获取给定日期开始时间

方法定义

```
def day_begin(day=None, year=None, month=None, days=None):
    """
    获取给定日期开始时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :param days:
```


Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
:return:
"""
```

示例

```
from jy_utils.date_utils import day_begin
day_begin()
### datetime.datetime(2021, 9, 23, 0, 0)
day_begin(year=2021, month=1, days=1)
### datetime.datetime(2021, 1, 1, 0, 0)
```

8.4.10. j. 获取给定日期结束时间

法定义

```
def day_end(day=None, year=None, month=None, days=None):
    """
    获取给定日期结束时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :param days:
    :return:
    """
```

例

```
from jy_utils.date_utils import day_end
day_end()
### datetime.datetime(2021, 9, 23, 23, 59, 59)
day_end(year=2021, month=1, days=1)
### datetime.datetime(2021, 1, 1, 23, 59, 59)
```

8.4.11. k. 获取给定日期昨天开始时间

法定义

```
def yesterday_begin(day=None, year=None, month=None, days=None):
    """
    获取给定日期昨天开始时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :param days:
    :return:
    """
```

示例

```
from jy_utils.date_utils import yesterday_begin
yesterday_begin()
### datetime.datetime(2021, 9, 22, 0, 0)
yesterday_begin(year=2021, month=1, days=1)
### datetime.datetime(2020, 12, 31, 0, 0)
```

1.8.4.12. l. 获取给定日期昨天结束时间

方法定义

```
def yesterday_end(day=None, year=None, month=None, days=None):
    """
    获取给定日期昨天结束时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :param days: 加减天数
    :return:
    """
```

示例

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法库
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
from jy_utils.date_utils import yesterday_end
yesterday_end()
### datetime.datetime(2021, 9, 22, 23, 59, 59)
yesterday_end(year=2021, month=1, days=1)
### datetime.datetime(2020, 12, 31, 23, 59, 59)
```

8.4.13. m. 获取给定日期月初时间

法定义

```
def month_begin(day=None, year=None, month=None, months=0):
    """
    获取给定日期月初时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :param months:
    :return:
    """
```

例

```
from jy_utils.date_utils import month_begin
month_begin()
### datetime.datetime(2021, 9, 1, 0, 0)
month_begin(year=2021, month=1)
### datetime.datetime(2021, 1, 1, 0, 0)
month_begin(months=-3)
### datetime.datetime(2021, 6, 1, 0, 0)
month_begin(year=2021, month=1, months=1)
### datetime.datetime(2021, 2, 1, 0, 0)
```

8.4.14. n. 获取给定日期月末时间

法定义

```
def month_end(day=None, year=None, month=None, months=0):
    """
    获取给定日期月末时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :param months:
    :return:
    """
```

示例

```
from jy_utils.date_utils import month_end
month_end()
### datetime.datetime(2021, 9, 30, 23, 59, 59)
month_end(year=2021, month=1)
### datetime.datetime(2021, 1, 31, 23, 59, 59)
month_end(months=-1)
### datetime.datetime(2021, 8, 31, 23, 59, 59)
month_end(year=2021, month=1, months=-1)
### datetime.datetime(2020, 12, 31, 23, 59, 59)
```

1.8.4.15. o. 获取给定日期上个月月初时间

方法定义

```
def last_month_begin(day=None, year=None, month=None):
    """
    获取给定日期上个月月初时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :return:
    """
```

示例

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法库
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
from jy_utils.date_utils import last_month_begin
last_month_begin()
### datetime.datetime(2021, 8, 1, 0, 0)
last_month_begin(year=2021, month=1)
### datetime.datetime(2020, 12, 1, 0, 0)
```

8.4.16. p. 获取给定日期上个月月末时间

法定义

```
def last_month_end(day=None, year=None, month=None):
    """
    获取给定日期上个月月末时间
    :param day: 默认为当前时间
    :param year:
    :param month:
    :return:
    """
```

```
from jy_utils.date_utils import last_month_end
last_month_end()
### datetime.datetime(2021, 8, 30, 23, 59, 59)
last_month_end(year=2021, month=1)
### datetime.datetime(2020, 12, 31, 23, 59, 59)
```

1.8.4.17. q. 获取给定日期年初时间

方法定义

```
def year_begin(day=None, year=None):
    """
    获取给定日期年初时间
    :param day: 默认为当前时间
    :param year:
    :return:
    """
```

示例

```
from jy_utils.date_utils import year_begin
year_begin()
### datetime.datetime(2021, 1, 1, 0, 0)
year_begin(year=2020)
### datetime.datetime(2020, 1, 1, 0, 0)
```

1.8.4.18. r. 获取给定日期年末时间

方法定义

```
def year_end(day=None, year=None):
    """
    获取给定日期年末时间
    :param day: 默认为当前时间
    :param year:
    :return:
    """
```

示例

```
from jy_utils.date_utils import year_end
year_end()
### datetime.datetime(2021, 12, 31, 23, 59, 59)
year_end(year=2020)
### datetime.datetime(2020, 12, 31, 23, 59, 59)
```

1.8.4.19. s. 当前日期开始时间字符串

方法定义

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
def cur_day_begin_str():
    """当前日期开始时间字符串"""
```

1.8.4.20. t. 当前日期结束时间字符串

法定义

```
lef cur_day_end_str():
    """当前日期结束时间字符串"""
```

8.4.21. u. 获取给定时间的字符串

法定义

```
lef day_str(day, fmt):
    """
    获取给定时间的字符串
    :param day:
    :param fmt: 时间格式, 默认DATE_FMT_HMS
    :return:
    """
```

8.4.22. v. 获取当前日期开始时间

法定义

```
def cur_day_begin():
    """当前日期开始时间"""
```

8.4.23. w. 获取当前日期结束时间

法定义

```
lef cur_day_end():
    """当前日期结束时间"""
```

1.8.4.24. x. 昨天开始时间字符串

方法定义

```
def yesterday_str(fmt):
    """
    昨天开始时间字符串
    :param fmt: 格式, 默认DATE_FMT_HMS
    :return:
    """
```

1.8.4.25. y. 给定日期加days天

方法定义

```
def add_days(day, days):
    """
    加days天
    :param day:
    :param days:
    :return:
    """
```

示例

```
from jy_utils.date_utils import add_days, datetime
add_days(datetime(2021, 9, 1), 11)
### datetime.datetime(2021, 9, 12, 0, 0)
add_days(-3)
### datetime.datetime(2021, 9, 20, 14, 28, 32, 663742)
```

1.8.4.26. z. 给定日期加months月

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

方法定义

```
def add_months(day, months, p_day=0):
    """
    加monthss月
    :param day:
    :param months:
    :param p_day:
    :return:
    """
```

例

```
from jy_utils.date_utils import add_months, datetime
add_months(datetime(2021, 9, 1), 1)
### datetime.datetime(2021, 10, 1, 0, 0)
add_months(-3)
### datetime.datetime(2021, 6, 23, 14, 31, 14)
```

8.4.27. aa. 获取给定时间的月初时间

方法定义

```
def month_first(day, days=1):
    """
    给定时间的月初
    :param day:
    :param days: 日期偏移量
    :return:
    """
```

8.4.28. ab. 获取给定时间的月初字符串

方法定义

```
def month_first_str(day, fmt, hms=False):
    """
    给定时间的月初字符串
    :param day:
    :param fmt:
    :param hms:
    :return:
    """
```

1.8.4.29. ac. 获取给定时间的月末时间

方法定义

```
def month_last(day):
    """
    给定时间的月末
    :param day:
    :return:
    """
```

1.8.4.30. ad. 获取给定时间的月末字符串

方法定义

```
def month_last_str(day, fmt, hms=False):
    """
    给定时间的月末字符串
    :param day:
    :param fmt:
    :param hms:
    :return:
    """
```

1.8.4.31. ae. 获取给定日期当月的天数

方法定义

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法库
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
def month_days(day):
    """
    获取给定日期当月的天数
    :param day:
    :return: (当月第一天的星期序号[0:6], 当月总天数)
    """
```

例

```
from jy_utils.date_utils import month_days, datetime
month_days(datetime(2021, 8, 23))
## (6, 31)
```

8.4.32. af. 字符串转日期

法定义

```
def str_to_date(st, fmt):
    """
    字符串转日期
    :param st: 字符串
    :param fmt: 格式, 默认: DATE_FMT
    :return:
    """
```

8.4.33. ag. 判断给定的值是否为数字

方法定义

```
def is_number(val):
    """
    判断给定的值是否为数字
    :param val:
    :return:
    """
```

8.4.34. ah. 值转时间

方法定义

```
def convert_val_to_datetime(val, fmt):
    """
    值转时间
    :param val: 值
    :param fmt: 格式, 默认: DATE_FMT_HMS
    :return:
    """
```

1.8.5. 5. dict_utils: 字典类工具方法库

1.8.5.1. a.深度更新dict

方法定义

```
def deep_update(d1: dict, d2: dict, distinct=False):
    """
    深度更新dict, 如果源字典(d1)key对应的值是集合类对象, 则如果d2存在相同的key, 则合并d1[v]和d2[v], 而不覆盖d1[v]
    :param d1: 源dict
    :param d2:
    :param distinct: 合并集合对象是否去重, 默认不去重
    :return: 更新后的源dict, 即d1
    """
```

示例

1.8.5.2. b.左保全合并dict

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

方法定义

```
def left_union(d1: dict, d2: dict):
    """
    左保全合并dict
    :param d1: 源dict
    :param d2:
    :return: 更新后的源dict, 即d1
    """
```

例

1.8.6. 6. excel_utils: excel文件类工具方法库

1.8.7. 7.file_utils: 文件类工具方法库

1.8.7.1. a.从文本文件中读取字符串

方法定义

```
def read_str(path, size=-1, mode='r', encoding='utf-8'):
    """
    从文本文件中读取字符串
    :param path: 文本文件路径
    :param size: 读取大小, -1: 全量读取
    :param mode: 读取模式
    :param encoding: 文件编码
    :return: str
    """
```

1.8.7.2. b.从文件中读取每行内容, return list

方法定义

```
def read_lines(path, mode='r', encoding='utf-8'):
    """
    从文件中读取每行内容, return list
    :param path: 文本文件路径
    :param mode: 读取模式
    :param encoding: 文件编码
    :return: list
    """
```

1.8.7.3. c.获取文件指定行内容

方法定义

```
def get_line(path, lineno, module_globals=None):
    """
    Get the lines for a Python source file from the cache.
    :param path: 文件路径
    :param lineno: 行号
    :param module_globals:
    :return:
    """
```

1.8.7.4. d.分批从大文件中读取

方法定义

```
def read_block(path, size=-1, mode='r', encoding='utf-8'):
    """
    分批从大文件中读取
    :param path: 文件路径
    :param size: 每次读取大小
    :param mode: 模式
    :param encoding: 编码
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法库
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
:return: generator
"""
```

1.8.7.5. e.按行读取大文件，返回生成器

法定义

```
def read_line(path, mode='r', encoding='utf-8'):
    """
    按行读取大文件
    :param path: 文件路径
    :param mode: 模式
    :param encoding: 编码
    :return: generator
    """
```

8.7.6. f.写入数据到文件

法定义

```
def write_file(path, data: str, mode='a+', encoding='utf-8'):
    """
    写入数据到文件
    :param path: 写入路径
    :param data: 待写入数据
    :param mode: 文件打开模式
    :param encoding: 文件编码
    :return:
    """
```

8.7.7. g.写入多行数据到文件

法定义

```
def write_file_by_lines(path, lines, mode='a+', encoding='utf-8'):
    """
    写入多行数据到文件
    :param path: 写入路径
    :param lines: 待写入数据
    :param mode: 文件打开模式
    :param encoding: 文件编码
    :return:
    """
```

1.8.7.8. h.检查数据是否有换行符，如果没有则添加

方法定义

```
def check_linesep(line: str):
    """
    检查数据是否有换行符，如果没有则添加
    :param line:
    :return:
    """
```

1.8.7.9. i.移除换行符

方法定义

```
def rmv_linesep(line: str, only_tail=1):
    """
    移除末尾的换行符
    :param line: 数据行
    :param only_tail: 只移除末尾
    :return:
    """
```

1.8.8. 8. func_utils:

1.8.8.1. a.注册方法

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle/MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

方法定义

```
def regist_func(name=None, app=None, level=1, default_params=None, replace=False):
    """
    regist func for later use
    :param name:
    :param app:
    :param level:
    :param default_params:
    :param replace:
    :return:
    """
```

例

```
from jy_utils.func_utils import regist_func

## 定义执行方法
@regist_func(name="", app=app, )
def method(*args, **kw):
    pass
```

8.8.2. b.执行任务

```
def run_task(app, log_model=None):
    """
    执行任务
    :param app: 任务所在app
    :param log_model: 执行日志数据模型
    :return:
    """
```

```
from jy_utils.func_utils import run_task

## 定义执行方法
@run_task(app=app, log_model=None)
def task(*args, **kw):
    pass
```

1.8.8.3. c.重试

方法定义

```
def retry(wait_rule='5:3:15', split='', def_result=None):
    """
    重试
    :param wait_rule: 重试等待时间规则 '%d:%d:%d' % (最多重试次数, 等待时间, 最大超时时间)
    :param split:
    :param def_result:
    :return:
    """
```

示例

```
from jy_utils.func_utils import run_task

## 定义执行方法
@run_task(app=app, log_model=None)
def task(*args, **kw):
    pass
```

1.8.9. 9. gps_utils:

1.8.9.1. a.gps坐标转换类

方法列表

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle/MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
def gcj02_to_bd09(self, gcj_lng, gcj_lat):
    """
    实现GCJ02向BD09坐标系的转换
    :param lng: GCJ02坐标系下的经度
    :param lat: GCJ02坐标系下的纬度
    :return: 转换后的BD09下经纬度
    """

    def bd09_to_gcj02(self, bd_lng, bd_lat):
        """
        实现BD09坐标系向GCJ02坐标系的转换
        :param bd_lng: BD09坐标系下的经度
        :param bd_lat: BD09坐标系下的纬度
        :return: 转换后的GCJ02下经纬度
        """

    def wgs84_to_gcj02(self, lng, lat):
        """
        实现WGS84坐标系向GCJ02坐标系的转换
        :param lng: WGS84坐标系下的经度
        :param lat: WGS84坐标系下的纬度
        :return: 转换后的GCJ02下经纬度
        """

    def gcj02_to_wgs84(self, gcj_lng, gcj_lat):
        """
        实现GCJ02坐标系向WGS84坐标系的转换
        :param gcj_lng: GCJ02坐标系下的经度
        :param gcj_lat: GCJ02坐标系下的纬度
        :return: 转换后的WGS84下经纬度
        """

    def bd09_to_wgs84(self, bd_lng, bd_lat):
        """
        实现BD09坐标系向WGS84坐标系的转换
        :param bd_lng: BD09坐标系下的经度
        :param bd_lat: BD09坐标系下的纬度
        :return: 转换后的WGS84下经纬度
        """

    def wgs84_to_bd09(self, lng, lat):
        """
        实现WGS84坐标系向BD09坐标系的转换
        :param lng: WGS84坐标系下的经度
        :param lat: WGS84坐标系下的纬度
        :return: 转换后的BD09下经纬度
        """

    def wgs84_to_webmercator(self, lng, lat):
        """
        实现WGS84向web墨卡托的转换
        :param lng: WGS84经度
        :param lat: WGS84纬度
        :return: 转换后的web墨卡托坐标
        """

    def webmercator_to_wgs84(self, x, y):
        """
        实现web墨卡托向WGS84的转换
        :param x: web墨卡托x坐标
        :param y: web墨卡托y坐标
        :return: 转换后的WGS84经纬度
        """
```

示例

1.8.10. 10. http_utils: 网络连接类工具方法库

1.8.10.1. a. 邮件通知

```
from jy_utils.http_utils import jy_mail

# 定义执行方法
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
"""
jy_mail 参数:
subject: 邮件主题, 不传则默认为: 监控情况
_from: 发送人, 不传则默认为jyf_jk@jiayuan.com
passwd: 发送人邮箱密码, 默认发送人(jyf_jk@jiayuan.com)不需要传此参数
_to: 接收人, 多个接收人以','分割
file_name: 附件名字, 如果不需要发送附件则不传
file_path: 附件对应的文件路径, 如果不需要发送附件则不传
"""

jy_mail(subject='test', _to='zhangjiawen@jiayuan.com', file_name='test', file_path='/u01/work/

def test():
    result = ''
    result = 'test' ### 方法执行内容

    # 执行结果写入文件
    with open('/u01/work/jiayuan/test.txt', 'w') as f:
        f.write(result)

    # 返回执行结果, 执行结果自动邮件发出
    return result
```

8.10.2. b. 微信通知

```
from jy_utils.http_utils import weixin
# 定义执行方法
"""
weixin参数:
user: 消息接收人(企业微信账号), 通知多人以'|'分割
"""
def weixin('zhangjiawen'):
def test():
    result = ''
    result = 'test' ### 方法执行内容

    # 返回执行结果, 执行结果自动企业微信通知接收人
    return result
```



1.8.11. 11. instance_utils:

1.8.11.1. a. 获取实例

方法定义

```
def instance(cls):
    """
    获取实例
    :param cls: 实例类型
    :return:
    """
```

示例

1.8.12. 12. json_utils:

1.8.12.1. a. 对象转json字符串

方法定义

```
def dumps(item):
    """
    对象转json字符串
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
:param item: 对象
:return:
"""
```

示例

8.12.2. b.json字符串转对象

法定义

```
def loads(json_str, encoding='utf8', **kw):
    """
    json字符串转对象
    :param json_str:
    :param encoding:
    :param kw:
    :return:
    """
```

..8.12.3. c.从jsonfile加载对象

方法定义

```
def load_json_file(path, encoding='utf8', **kw):
    """
    从jsonfile加载对象
    :param path:
    :param encoding:
    :param kw:
    :return:
    """
```

示例

1.8.13. 13. list_utils:

1.8.13.1. a.获取对象集合中符合条件的第一个对象

方法定义

```
def first(items, condition=None, def_val=None):
    """
    获取对象集合中符合条件的第一个对象
    :param items: 对象集合
    :param condition: 过滤条件
    :param def_val: 默认返回值
    :return:
    """
```

示例

1.8.13.2. b.list/tuple转set

方法定义

```
def to_set(a):
    """
    list/tuple转set
    :param a:
```

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
:return:
"""
```

示例

8.13.3. c.list/tuple/set 求并集

法定义

```
def union(a, b, distinct=False):
    """
    list/tuple/set 求并集
    :param a: 第一个集合
    :param b: 第二个集合
    :param distinct: 结果是否去重
    :return:
    """
```

例

8.13.4. d.返回第n个元素

方法定义

```
def nrd(items, n, last_replace=False):
    """
    返回第n个元素
    :param items:
    :param n: 第n个
    :param last_replace: 如果元素个数不足是否用最后一个代替, 如果否, 元素不足时返回None
    :return:
    """
```

示例

1.8.14. 14. log_utils

1.8.14.1. a.输出日志

方法定义

```
def log(app=None, msg='', level='info'):
    """
    输出日志
    :param app: 所在app(flask启动对象)
    :param msg: 日志内容
    :param level: 日志级别
    :return:
    """
```

1.8.15. 15. mobile_utils

1.8.15.1. a.随机生成1个手机号

方法定义

```
def random_mobile(pries=None):
    """
    随机生成1个手机号
    :param pries: 指定前缀list
    :return:
    """
```

Table of Content

1. jydata-common

1.1. 一、jy_code

1.2. 二、jy_conf 公共配置

1.3. 三、jy_crypt 加解密

1.3.1. 1. base_bit_num 基础数位

1.3.1.1. a. 类定义

1.3.1.2. b. 已定义对象

1.3.2. 2. bit_num 进制数

1.3.2.1. a. 类定义

1.3.3. 3. 佳缘线上加解密

1.3.3.1. a. 身份证号加密

1.3.3.2. b. 身份证号解密

1.3.3.3. c. 姓名加解密

1.3.4. 4. 佳缘vip加解密

1.3.4.1. a. 佳缘vip数据加解密

1.4. 四、jy_db 数据库

1.4.1. 1. 获取数据库对象

1.4.1.1. a. 查看数据库配置信息

1.4.1.2. b. 获取数据库对象

1.4.2. 2. 数据库对象api

1.4.2.1. a. select: 查询, 适用于Oracle/MySQL

1.4.2.1.1. 方法定义

1.4.2.1.2. 示例

1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL

1.4.2.2.1. 方法定义

1.4.2.2.2. 示例

1.4.2.3. c. update_batch: 批量更新, 适用于MySQL

1.4.2.3.1. 方法定义

1.4.2.3.2. 示例

1.5. 五、jy_entity 实体基类

1.6. 六、jy_factory 工厂类

1.7. 七、jy_result 返回结果

1.8. 八、jy_util 工具/方法

1.8.1. 1. attr_utils: 对象属性工具方法库

1.8.1.1. a. 设置对象属性

1.8.2. 2. calender_utils: 日历工具方法库

1.8.2.1. a. 获取给定日期所在月份的工作日

1.8.3. 3. condition_utils: 条件类工具方法库

1.8.3.1. a. 常量定义

1.8.3.2. b. 格式化时间元素数字

1.8.3.3. c. 时间戳转datetime

示例

8.15.2. b.随机生成n个手机号

法定义

```
def random_mobiles(n, pries=None):
    """
    随机生成n个手机号
    :param n: 生成手机号数量
    :param pries: 指定前缀list
    :return: 包含n个手机号的生成器
    """
```

例

8.16. 16. model_utils

1.8.17. 17. num_utils

1.8.17.1. a.常量定义

```
MAX_INT = 2**32-1 # 最大整数
```

法定义

1.8.18. 18. obj_utils

1.8.18.1. a.根据type_name获取type

方法定义

```
def name2class(name: str):
    """
    根据type_name获取type
    :param name: type_name
    """
```

示例

1.8.18.2. b.对象序列化key

方法定义

```
def serializable_key(key, add_cls=False, cls_k='class'):
    """
    序列化key
    :param key:
    :param add_cls:
    :param cls_k:
    :return:
    """
```

示例

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1.8.18.3. c.对象转dict

法定义

```
def obj2dict(o, add_cls=False, cls_k='class'):
    """
    对象转dict
    :param o: 待转对象
    :param add_cls: 添加类型
    :param cls_k: 结果中类型字段对应的k值, 仅在默认('cls')与代转对象属性冲突时使用
    :return:
    """
```

例

8.18.4. d.dict 转对象

法定义

```
def dict2obj(d: dict, cls_k='class', copy=False, fields_dict=None):
    """
    dict 转对象
    :param d: 待转dict
    :param cls_k: 类型字段对应的k值
    :param copy: 是否全复制对象
    :param fields_dict: 参与创建对象的字段列表{cls: {dict_key: obj_param_name}}
    :return:
    """
```

例

1.8.18.5. e.如果对象为None则转为默认值

方法定义

```
def none(obj, v):
    """
    如果对象为None则转为默认值
    :param obj: 待转对象
    :param v: 默认值
    :return:
    """
```

示例

1.8.18.6. f.下一个值

方法定义

```
def next_val(obj, sort=False, reverse=False, limit=10000, def_val=None, params=None):
    """
    下一个值
    :param obj:
    :param sort:
    :param reverse:
    :param limit:
    :param def_val:
    :param params:
    :return:
    """
```

示例

Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

1.8.19. 19. operation_utils

8.19.1. a.追加/合并对象到list/dict

法定义

```
def add(o1, o2):  
    """  
    追加/合并对象到list/dict  
    :param o1:  
    :param o2:  
    :return:  
    """
```

8.20. 20. param_utils

8.20.1. a.生成任务参数

法定义

```
def gen_task_params(app, **kwargs):  
    """  
    生成任务参数  
    :param app:  
    :param kwargs:  
    :return:  
    """
```

8.20.2. b.处理接收参数

法定义

```
def handle_receive_args(args):  
    """  
    处理接收参数  
    :param args:  
    :return:  
    """
```

1.8.21. 21. str_utils

1.8.21.1. a.打包字符串为dict

方法定义

```
def zip_str(s: str):  
    """  
    将给定字符串打包成dict(), k, v分别为未知索引和对应的字符  
    :param s:  
    :return:  
    """
```

示例

```
from jy_utils import zip_str  
zip_str('123')  
### {0: '1', 1: '2', 2: '3'}
```

1.8.21.2. b.字符串内部排序

方法定义

```
def sorted_str(s: str, idxes=None, reverse=False):  
    """  
    字符串内部排序, 返回排序后的字符串
```


Table of Content

- 1. jydata-common
 - 1.1. 一、jy_code
 - 1.2. 二、jy_conf 公共配置
 - 1.3. 三、jy_crypt 加解密
 - 1.3.1. 1. base_bit_num 基础数位
 - 1.3.1.1. a. 类定义
 - 1.3.1.2. b. 已定义对象
 - 1.3.2. 2. bit_num 进制数
 - 1.3.2.1. a. 类定义
 - 1.3.3. 3. 佳缘线上加解密
 - 1.3.3.1. a. 身份证号加密
 - 1.3.3.2. b. 身份证号解密
 - 1.3.3.3. c. 姓名加解密
 - 1.3.4. 4. 佳缘vip加解密
 - 1.3.4.1. a. 佳缘vip数据加解密
 - 1.4. 四、jy_db 数据库
 - 1.4.1. 1. 获取数据库对象
 - 1.4.1.1. a. 查看数据库配置信息
 - 1.4.1.2. b. 获取数据库对象
 - 1.4.2. 2. 数据库对象api
 - 1.4.2.1. a. select: 查询, 适用于Oracle/MySQL
 - 1.4.2.1.1. 方法定义
 - 1.4.2.1.2. 示例
 - 1.4.2.2. b. update: 增删改, 适用于Oracle/MySQL
 - 1.4.2.2.1. 方法定义
 - 1.4.2.2.2. 示例
 - 1.4.2.3. c. update_batch: 批量更新, 适用于Oracle/MySQL
 - 1.4.2.3.1. 方法定义
 - 1.4.2.3.2. 示例
 - 1.5. 五、jy_entity 实体基类
 - 1.6. 六、jy_factory 工厂类
 - 1.7. 七、jy_result 返回结果
 - 1.8. 八、jy_util 工具/方法
 - 1.8.1. 1. attr_utils: 对象属性工具方法库
 - 1.8.1.1. a. 设置对象属性
 - 1.8.2. 2. calender_utils: 日历工具方法库
 - 1.8.2.1. a. 获取给定日期所在月份的工作日
 - 1.8.3. 3. condition_utils: 条件类工具方法库
 - 1.8.4. 4. date_utils: 日期时间类工具方法库
 - 1.8.4.1. a. 常量定义
 - 1.8.4.2. b. 格式化时间元素数字
 - 1.8.4.3. c. 时间戳转datetime

```
:param s:
:param idxes: 给定的索引顺序
:param reverse: 是否反转
:return:
"""
```

例

```
from jy_utils import sorted_str
sorted_str('1d3a2d45')
## '12345add'
sorted_str('1d3a2d45', reverse=True)
## 'dda54321'
sorted_str('1d3a2d45', idxes='1042')
## 'd123'
sorted_str('1d3a2d45', idxes='1042', reverse=True)
## '321d'
```

8.21.3. c.有分隔符的字符串转换成dict

法定义

```
def str2dict(s: str, sps: list, keys: list = None) -> dict:
    """
    把有分隔符的字符串转换成dict
    :param s: 输入字符串
    :param sps: 分隔符
    :param keys: 给定key集合, 在没有二级分隔符(sp)时, 将一级分割结果依次匹配key
    :return: 返回分割成的dict
    """
```

8.21.4. d.字符串转值list

方法定义

```
def str2dict(s: str, sps: list, keys: list = None) -> dict:
    """
    把有分隔符的字符串转换成dict
    :param s: 输入字符串
    :param sps: 分隔符
    :param keys: 给定key集合, 在没有二级分隔符(sp)时, 将一级分割结果依次匹配key
    :return: 返回分割成的dict
    """
```

示例

1.8.21.5. e.根据给定字符集截取字符串

方法定义

```
def str2dict(s: str, sps: list, keys: list = None) -> dict:
    """
    把有分隔符的字符串转换成dict
    :param s: 输入字符串
    :param sps: 分隔符
    :param keys: 给定key集合, 在没有二级分隔符(sp)时, 将一级分割结果依次匹配key
    :return: 返回分割成的dict
    """
```

示例