

**P.409**

以下3つの関数を左端まで移動した上で、各インデントを4スペースに修正

- api\_search
- parse\_listings
- process\_listings

**修正前**

```
def api_search(address, site='yellowpages.com'):
    """ あるサイトのビジネスアドレスを受け取り,APIを介してその戻り値を返す """
    req_params = {}
    req_params.update({
        'key': get_api_key(site),
        'term': address['name'],
        'searchloc': '{0}, {1}, {1}'.format(address['street'],
                                           address['city'],
                                           address['state'])})
    return requests.post(search_api % locals(), params=req_params)

def parse_listings(addresses, sites):
    """ アドレスのリストを受け取り,各ビジネスリスティングをフェッチ """
    for site in sites:
        for address in addresses:
            listing = api_search(address, site)
            # ビジネスリスティングを処理
            process_listing(listing, site)

def process_listings(listing, site):
    """ ビジネスリスティングを処理 """
    # 何らかの重たい処理
    # 処理内容そのものは重要でないため省略
```

**修正後**

```

def api_search(address, site='yellowpages.com'):
    """ あるサイトのビジネスアドレスを受け取り,APIを介してその戻り値を返す """
    req_params = {}
    req_params.update({
        'key': get_api_key(site),
        'term': address['name'],
        'searchloc': '{0}, {1}, {1}'.format(address['street'],
                                           address['city'],
                                           address['state'])})
    return requests.post(search_api % locals(), params=req_params)

def parse_listings(addresses, sites):
    """ アドレスのリストを受け取り,各ビジネスリスティングをフェッチ """
    for site in sites:
        for address in addresses:
            listing = api_search(address, site)
            # ビジネスリスティングを処理
            process_listing(listing, site)

def process_listings(listing, site):
    """ ビジネスリスティングを処理 """
    # 何らかの重たい処理
    # 処理内容そのものは重要でないため省略

```

**P.411**

下から5行目のreq\_paramsを左に1スペース移動したのち整形

修正前

```

req_params = {}
req_params.update({
    'key': get_api_key(site), 'term': address['name'],
    'searchloc': '{0}, {1}, {1}'.format(address['street'],
                                        address['city'], address['state'])})

```

修正後

```

req_params = {}
req_params.update({
    'key': get_api_key(site),
    'term': address['name'],
    'searchloc': '{0}, {1}, {1}'.format(address['street'],
                                        address['city'],
                                        address['state'])})

```

**P.412**

下から5行目～2行目までを左に1スペース移動

修正前

```

if cached_data != None:
    print('=>from cache<=')
    return json.loads(cached_data)
# 登録されていなければ関数を呼び出し,またTTLを設定しておく
result = func(*args, **kwargs)
cache.set(key, json.dumps(result), ttl)
return result

```

**修正後**

```

if cached_data != None:
    print('=>from cache<=')
    return json.loads(cached_data)
# 登録されていなければ関数を呼び出し,またTTLを設定しておく
result = func(*args, **kwargs)
cache.set(key, json.dumps(result), ttl)
return result

```

**P.416**

上から5・6行目のインデントずれを修正

**修正前**

```

def __init__(self, **kwargs):
    kwargs['choices'] = ['male', 'female']
kwargs['default'] = 'male'
BaseType.__init__(self, **kwargs)

```

**修正後**

```

def __init__(self, **kwargs):
    kwargs['choices'] = ['male', 'female']
    kwargs['default'] = 'male'
    BaseType.__init__(self, **kwargs)

```

**P.422**

上から3行目のインデントずれを修正

**修正前**

```

    formatter = logging.Formatter('%(asctime)s [%(timespent)s]: %
(levelname)-8s -
                                     %(message)s', datefmt='%Y-%m-%d
%H:%M:%S')
    handle.setFormatter(formatter)

```

**修正後**

```

    formatter = logging.Formatter('%(asctime)s [%(timespent)s]: %
(levelname)-8s -
                                     %(message)s', datefmt='%Y-%m-%d
%H:%M:%S')
    handle.setFormatter(formatter)

```