

# CALLBACK В АСИНХРОННЫХ МЕТОДАХ

## ВВЕДЕНИЕ

### Задачи урока

**01** Как создавать callback для асинхронных методов

**02** Как вызывать callback в асинхронных функциях

## Callback

В асинхронных функциях можно добавлять **callback**

```
...  
  
import time  
from turtle import pen  
  
import asyncio  
import functools  
  
def upper(future):  
    print(future.result().upper())  
  
def save_to_db(site, future):  
    print("db", site, future.result())  
  
def print_space(site, future):  
    print("\n")  
  
def save_to_minio(site, future):  
    print("minio", site, future.result())  
  
async def parser(site_name, future):  
    await asyncio.sleep(2)  
    print(site_name)  
    future.set_result(f"data from {site_name}")  
  
def get_callback(site):  
    future = asyncio.Future()  
    for callback in [save_to_db, save_to_minio, print_space]:  
        future.add_done_callback(functools.partial(callback, site))  
    return future
```

```
if __name__ == '__main__':
    start_time = time.time()

    jobs = []
    for site in ["site 1", "site 2", "site 3"]:
        jobs.append(asyncio.ensure_future(parser(site,
get_callback(site))))
    event_loop = asyncio.get_event_loop()
    event_loop.run_until_complete(asyncio.gather(*jobs))
    event_loop.close()

    print("--- %s seconds ---" % (time.time() - start_time))
```

Создаем **Callback** из функций **save\_to\_db**, **save\_to\_minio**, **print\_space**

```
...
def get_callback(site):
    future = asyncio.Future()
    for callback in [save_to_db, save_to_minio, print_space]:
        future.add_done_callback(funcutils.partial(callback, site))
    return future
```

Далее создаем задачи которые будут выполняться асинхронно и передаем **callback**

```
...
jobs = []
for site in ["site 1", "site 2", "site 3"]:
    jobs.append(asyncio.ensure_future(parser(site,
get_callback(site))))
```

И вызываем **callback** в функции

```
...
async def parser(site_name, future):
    await asyncio.sleep(2)
    print(site_name)
    future.set_result(f"data from {site_name}")
```

## ИТОГИ

- ✓ При необходимости обрабатывать данные в асинхронных методах, как только эти данные были получены, можно использовать **callback**
  - ✓ **Callback'ов** может быть много
- 