

# Cairo Crash Course

---

## Advanced Data Types - Exercise 7

In this exercises we will implement a smart contract that manages allowlist of tokens.

### Contract Implementation

In `advanced_types_7.cairo` implement the contract `SimpleMap` according to the following instructions:

1. Define the following contract storage:

```
tokens: LegacyMap::<felt252, bool>
```

2. Implement the functions defined in the interface `ISimpleMap`:

```
ISimpleMap<TContractState> {  
    fn add_token(ref self: TContractState, token_name: felt252);  
    fn is_token_in_storage(self: @TContractState, key: felt252) -> bool;  
}
```

- `add_token()` - Add a token to the storage
- `is_token_in_storage()` - Check if a token is in the storage

### Writing Tests

In the test block `test_simple_map` in the file `advanced_types_7_test.cairo`, write the following tests:

1. Declare and deploy the contract `SimpleMap` and create a dispatcher for it.
2. Add the following tokens using the `add_token` function, and for every token, make sure it is in the storage using the `is_token_in_storage` function:
  - `ETH`
  - `STRK`
  - `USDC`
  - `DOGE`