

# ERC20 Tokens - allowed, transferFrom(...) and approve(...)

---

- **transfer()** function is used for sending tokens from one user to another, but it doesn't work well when tokens are being used to pay for a function in a smart contract;
- ERC20 standard defines a mapping data structure named **allowed** and 2 functions **approve(...)** and **transferFrom(...)** that permit a token owner to **give another address approval to transfer up to a no. of tokens** known as **allowance**.

Allowances for an address can only be set by the owner of that address, ensuring that no-one can use the allowance feature to steal tokens.

# ERC20 Token - allowed, transferFrom(...) and approve(...)

---

Let's imagine there are 2 users **A** and **B**. **A** has **1000** tokens and wants to give **permission to B**, which in most cases is a contract, to **spend 100 of them** or in other words to give permission to B to transfer 100 tokens from A account to his account.

1. A, the owner, will call `approve(address_of_B, 100)`. After that the `allowed` data structure will contain the following information:  
`allowed[address_of_A][address_of_B] = 100`
2. If B wants to later transfer 20 tokens from A to his account, B will execute the `transferFrom()` function in this way: `transferFrom(address_of_A, address_of_B, 20)`.

After calling the `transferFrom()` function (by B) the balance of A decreased with 20 and the balance of B increased with 20 tokens and the allowed data structure will contain the following info: `allowed[address_of_A][address_of_B] = 80`