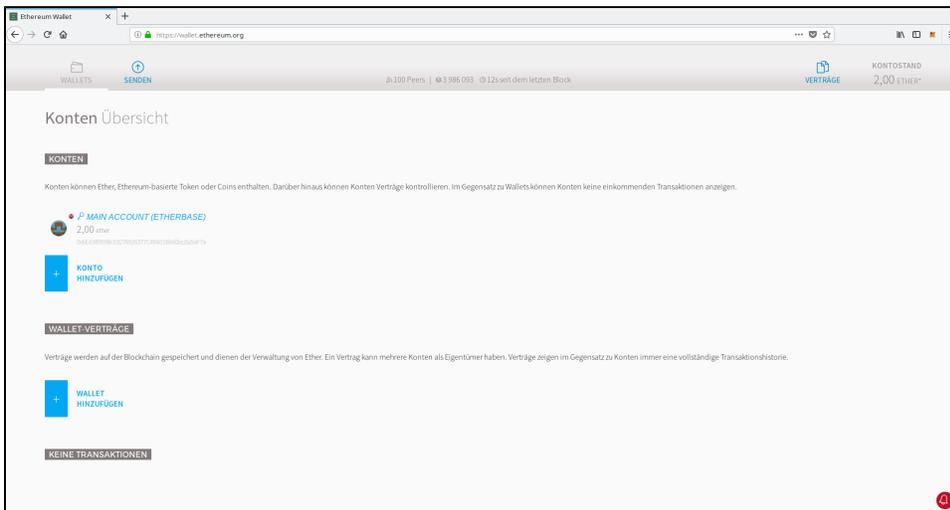


Praxis session 2b - ERC20 Token

ERC20 Token

In der nächsten Session sollen Sie eine der erfolgreichsten Blockchainanwendungen kennenlernen, die Ethereum-basierten Token. Die meisten von Ihnen beruhen auf einem Quasi-Standard, der unter dem Namen ERC20 (Ethereum Request for Comments) bekannt ist <https://github.com/ethereum/eips/issues/20>.

- Zunächst wollen wir Ihnen ein paar Einheiten des begehrten "**Competence Token**" zukommen lassen. Dazu verwenden wir ein "OnlineWallet" welches sich mit MetaMask verbindet.
- Öffnen Sie den Firefox Browser und rufen Sie die folgende Seite auf <https://wallet.ethereum.org>
- Melden Sie sich nun bei Ihrem **MetaMask** Addon (Fuchs oben rechts) mit Ihrem Passwort an und aktualisieren Sie die Seite. Sollten Sie das Addon noch nicht installiert haben, führen Sie die entsprechenden Schritte aus der 2. Session durch.
- Wählen Sie in MetaMask das **Ropsten Testnet** aus.
- Jetzt müssen wir dem Wallet den Zugriff auf MetaMask erlauben. Klicken Sie auf den **Fuchs**, dann auf das **Account-Symbol (bunter Kreis oben rechts)**, dann **Settings**.
 - Wählen Sie "**Connections**" aus.
 - Tragen Sie unter **Add Site** folgende Adresse ein: wallet.ethereum.org. Klicken Sie auf **Add**.
- Nehmen Sie den Warnhinweis der Wallet zur Kenntnis.



- Im nächsten Schritt sollen Sie den "Competence Token" zu Ihrer Wallet hinzufügen.
- Klicken Sie dazu oben rechts auf **VERTRÄGE**.
- Klicken Sie nun unten links auf **TOKEN HINZUFÜGEN**.
- Geben Sie in der Vertragsadresse: **0x1225446678163b14859eed119831a68576fc15ae** ein und bestätigen Sie mit **OK**.

Token hinzufügen

TOKEN VERTRAGSADRESSE

 0x1225446678163b14859eed119831a685

TOKENNAME

Competence Token

TOKENSYMBOL

CT

NACHKOMMASTELLEN DER KLEINSTEN EINHEIT

0

VORSCHAU



ABBRECHEN
OK

EIGENE TOKEN

Token sind Währungen oder Tauschgegenstände, die auf der Ethereum-Plattform aufsetzen. Damit ein Konto ein Token beobachten Sie z.B. folgendes [Beispiel eines Token Vertrags](#) anpassen, oder mehr über [Ethereum Token](#) lernen.



+

TOKEN

HINZUFÜGEN

- Wenn Sie dem Dozenten in der letzten Session Ihre Adresse schon mitgeteilt haben, sollten Sie bereits ein paar CTs besitzen anderenfalls können Sie das auch heute noch via "collabedit" nachholen.
- Schauen Sie sich nun den "Competence Token" auch auf Etherscan an.

<https://ropsten.etherscan.io/token/0x1225446678163b14859eed119831a68576fc15ae>

Der eigene Token

- Nun sollen Sie Ihren eigenen Token ins Leben rufen.
- Öffnen Sie dazu im Firefox Browser die Remix IDE. Achten Sie darauf, dass Sie mit Meta Mask angemeldet sind.

<http://remix.ethereum.org>

- Erstellen Sie einen **neuen Smart Contract** über einen Klick auf das weiße Kreuz in schwarzem Kreis in der mittleren Spalte der Remix IDE und geben Sie als Namen "ERC20Token.sol" ein.
- Kopieren Sie sich das Grundgerüst in das Editorfenster.

Das Grundgerüst spiegelt die Funktionen wieder, die laut ERC20-Standard implementiert sein müssen. Beachten Sie, dass die Methoden hier noch leer sind und keine Funktion haben. Für die Implementierung der einzelnen Methoden sind die Entwickler verantwortlich.

```

pragma solidity ^0.5.11;
contract ERC20Token {

    event Transfer(address indexed _from, address indexed _to, uint _value);
    event Approval(address indexed _owner, address indexed _spender, uint _value);

    string public name;
    string public symbol;
    uint public totalSupply;
    mapping(address => uint) public balanceOf;
    mapping(address => mapping (address => uint256)) public allowance;

    constructor (uint _initialSupply, string memory _name, string memory _symbol) public
    {

    }

    function transfer(address _to, uint _value) public returns (bool success)
    {

    }

    function transferFrom(address _from, address _to, uint _value) public returns (bool success)
    {

    }

    function approve(address _spender, uint _value) public returns (bool success)
    {

    }
}

```

Erfüllen wir die einzelnen Methoden nun mit Leben.

1. Konstruktor

```

constructor (uint _initialSupply, string memory _name, string memory _symbol) public
{
    totalSupply = _initialSupply;
    name = _name;
    symbol = _symbol;
    balanceOf[msg.sender] = _initialSupply;
}

```

2. Transfer Methode

```

function transfer(address _to, uint _value) public returns (bool success)
{
    require(balanceOf[msg.sender] >= _value);
    require(balanceOf[_to] + _value > balanceOf[_to]);
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
    emit Transfer(msg.sender, _to, _value);
    return true;
}

```

3. Transfer From Methode

```

function transferFrom(address _from, address _to, uint _value) public returns (bool success)
{
    require(allowance[_from][msg.sender] >= _value);
    require(balanceOf[_from] >= _value);
    require(balanceOf[_to] + _value > balanceOf[_to]);
    allowance[_from][msg.sender] -= _value;
    balanceOf[_from] -= _value;
    balanceOf[_to] += _value;
    emit Transfer(_from, _to, _value);
    return true;
}

```

4. Approve Methode

```

function approve(address _spender, uint _value) public returns (bool success)
{
    allowance[msg.sender][_spender] = _value;
    emit Approval(msg.sender, _spender, _value);
    return true;
}

```

Token erstellen

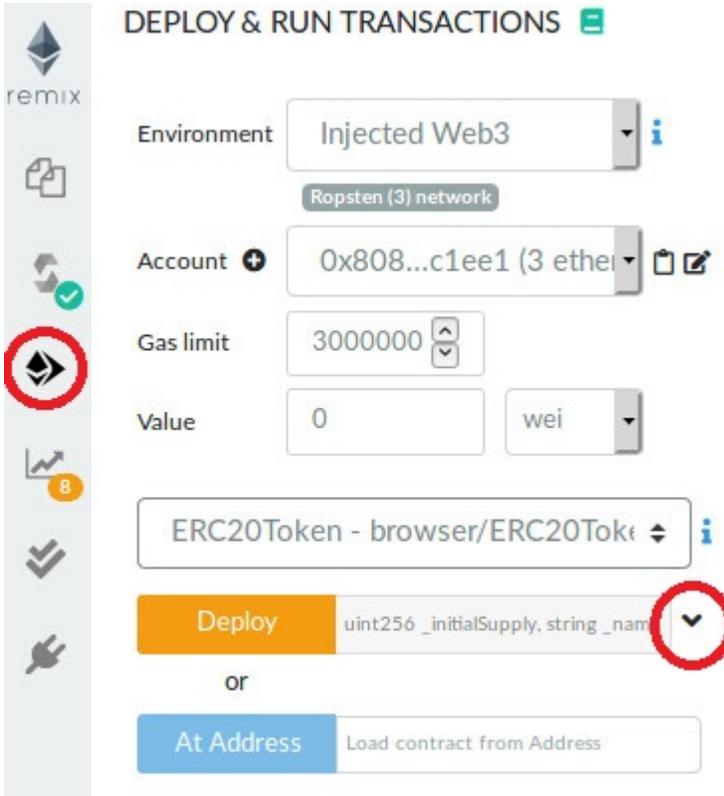
- Kompilieren Sie nun Ihren ERC20 Token. Gehen Sie dazu in der linken Spalte zu Solidity Compiler und klicken Sie auf "Compile ERC20Token".

The screenshot shows the Solidity Compiler interface in the Remix IDE. The left sidebar contains navigation icons, with the compiler icon circled in red. The main panel displays the following settings:

- Compiler:** 0.5.11+commit.c082d0 (dropdown menu)
- Language:** Solidity (dropdown menu)
- EVM Version:** compiler default (dropdown menu)
- Include nightly builds:**
- Compile ERC20Token.sol:** A dark blue button with a refresh icon, circled in red.
- Compiler Configuration:**
 - Auto compile
 - Enable optimization
 - Hide warnings
- No Contract Compiled Yet:** A yellow button at the bottom.

- Wenn beim Kompilieren keine Fehler aufgetreten sind, können wir den Contract deployen. Dazu müssen Sie zunächst in der linken Spalte der Remix IDE auf den Reiter "Deploy and Run transactions" wechseln.
- Vergewissern Sie sich, dass unter **Environment "Injected Web3"** sowie ein Account mit verfügbarem Rest-Ether ausgewählt ist.
- Bevor Sie auf "deploy" klicken, tragen Sie noch die Parameter Ihres Tokens (_initialSupply, _name, _symbol) ein. Klappen Sie dazu die Eingabefelder neben Deploy aus.

- Klicken sie anschließend auf "transact"

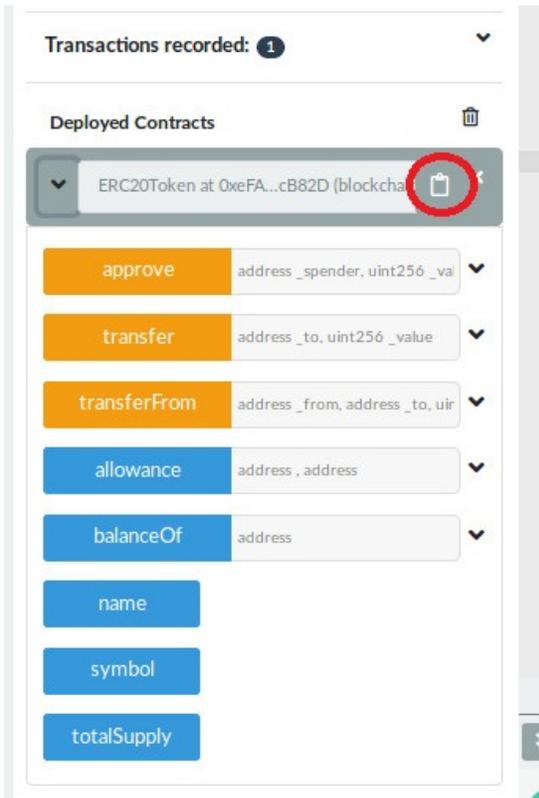


- Nachdem Sie auf "deploy" geklickt haben, müssen Sie die Transaktion mit Meta Mask noch bestätigen. (Sollte sich kein Popup öffnen, klicken Sie auf den Fuchs)
- Warten Sie bis der Contract gemined wurde.



Token testen

- Nun können Sie anfangen Ihren Token zu verteilen. Wir wollen dazu das Wallet aus dem 1. Schritt benutzen.
- Gehen Sie wieder auf die Webseite <https://wallet.ethereum.org> und fügen Sie über **VERTRÄGE (Contracts) TOKEN HINZUFÜGEN (Watch Token)** Ihren Token hinzu. Die Adresse Ihres Tokens finden Sie in der Remix IDE unter "Deployed Contracts" und kann mit Klick auf das Dokumentsymbol kopiert werden.



- Wenn Sie alles richtig gemacht haben, sollte die Wallet nun allein anhand der Adresse den Token automatisch erkennen und den von Ihnen vergebenen Namen und das Symbol korrekt anzeigen.

Token hinzufügen

TOKEN VERTRAGSADRESSE

0x08f85f31e833971a42f4ad85f8f017e726

TOKENNAME

TOKENSYMBOL

NACHKOMMASTELLEN DER KLEINSTEIN EINHEIT

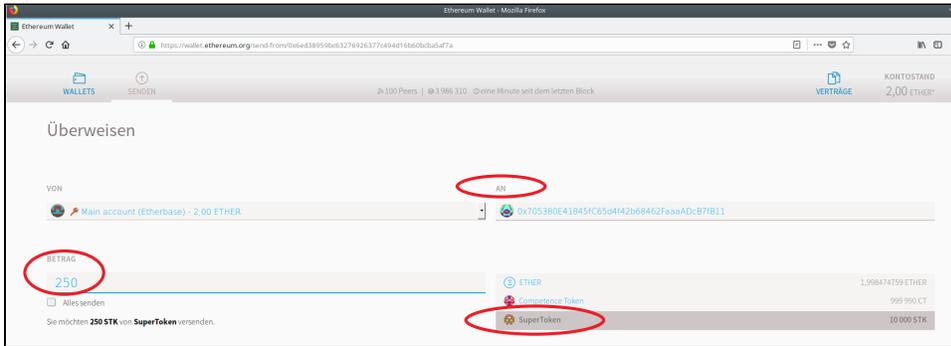
VORSCHAU

SUPERTOKEN
 0 STK
 0x08f85f31e833971a42f...

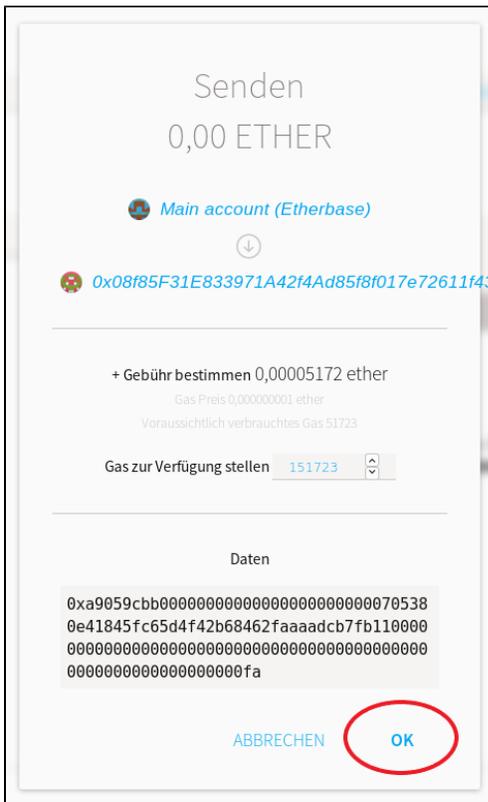
ABBRECHEN
OK

- Schicken Sie nun Ihren Mitstreitern ein paar Ihrer Token über das Wallet.
- Veröffentlichen Sie dazu zunächst die **Adresse Ihres Tokens** sowie die **Adresse Ihres eigenen Accounts**. Nutzen Sie dazu das "collabedit", das Ihnen der Dozent mitteilt,
- Suchen Sie sich aus der Liste nun eine Adresse eines Mitstreiters aus und Senden sie an diese Adresse ein paar Ihrer Token.

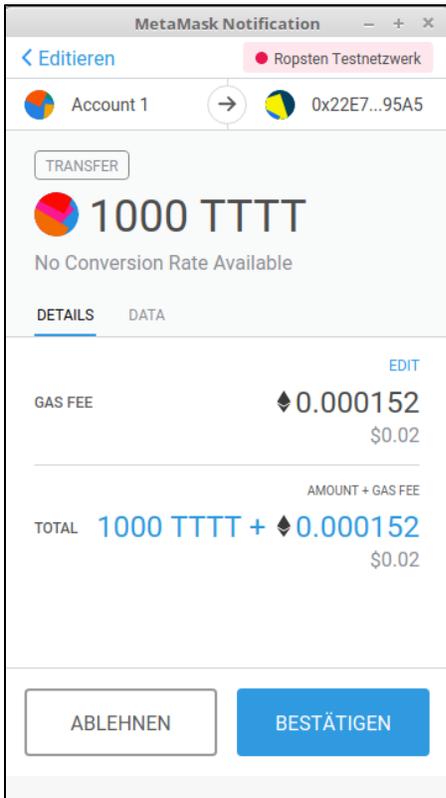
- Klicken Sie in der Wallet dazu auf "SENDEN", fügen sie in das "AN" Feld die Adresse des Empfängers ein.
- Wählen Sie einen Betrag und vergewissern Sie sich, dass sie Ihren Token als "Währung" ausgewählt haben.



- Scrollen Sie nun nach unten und klicken Sie auf **SENDEN**.
- Klicken Sie im nächsten Dialog auf "OK".



- Bestätigen Sie die Transaktion abschließend mit Meta Mask.



- Warten Sie bis Ihre Transaktion bestätigt wurde.
- Importieren Sie sich nun die Token Ihrer Mitstreiter aus dem "collabedit". Vielleicht hat Ihnen jemand ja ein paar Token gesendet.
- Abschließend können Sie sich Ihren eigenen Token auch auf Etherscan anschauen.

[https://ropsten.etherscan.io/token/ <adresse ihres tokens>](https://ropsten.etherscan.io/token/<adresse ihres tokens>)

Sie können Token auch zu Ihrem MetaMask Plugin hinzufügen.

- Klicken Sie dazu im MetaMask auf das **Burgermenü Symbol** und anschließend ganz unten auf "**ADD TOKEN**"
- Geben sie anschließend unter "**CUSTOM TOKEN**" die Adresse Ihres Tokens und des Competence Tokens ein.

🔍 ☆ 🦊

🦊 Ropsten Testnetzwerk 🌐

Token hinzufügen

Suche Custom Token

Token Contract Address

Tokensymbol [Editieren](#)

Dezimalangabe der Präzision